Results of the Ontology Alignment Evaluation Initiative 2012*

José Luis Aguirre[1], Kai Eckert[4], Jérôme Euzenat[1], Alfio Ferrara[2], Willem Robert van Hage[3], Laura Hollink[3], Christian Meilicke[4], Andriy Nikolov[5], Dominique Ritze[4], François Scharffe[6], Pavel Shvaiko[7], Ondřej Šváb-Zamazal[8], Cássia Trojahn[9], Ernesto Jimenez-Ruiz[11], Bernardo Cuenca Grau[11], and Benjamin Zapilko[10]

[1] INRIA & LIG, Montbonnot, France
{Jose-Luis.Aguirre,Jerome.Euzenat}@inria.fr
[2] Università degli studi di Milano, Italy
alfio.ferrara@unimi.it
[3] Vrije Universiteit Amsterdam, The Netherlands
{W.R.van.Hage,L.Hollink}@vu.nl
[4] University of Mannheim, Mannheim, Germany
{christian,dominique,kai}@informatik.uni-mannheim.de
[5] The Open University, Milton Keynes, United Kingdom
A.Nikolov@open.ac.uk
[6] LIRMM, Montpellier, France
francois.scharffe@lirmm.fr
[7] TasLab, Informatica Trentina, Trento, Italy
pavel.shvaiko@infotn.it
[8] University of Economics, Prague, Czech Republic
ondrej.zamazal@vse.cz
[9] IRIT – Université Toulouse II, Toulouse, France
cassia.trojahn@irit.fr
[10] GESIS – Leibniz Institute for the Social Sciences, Cologne, Germany
benjamin.zapilko@gesis.org
[11] University of Oxford, UK
{ernesto,berg}@cs.ox.ac.uk

Abstract. Ontology matching consists of finding correspondences between semantically related entities of two ontologies. OAEI campaigns aim at comparing ontology matching systems on precisely defined test cases. These test cases can use ontologies of different nature (from simple thesauri to expressive OWL ontologies) and use different modalities, e.g., blind evaluation, open evaluation, consensus. OAEI 2012 offered 7 tracks with 9 test cases followed by 21 participants. Since 2010, the campaign has been using a new evaluation modality which provides more automation to the evaluation. This paper is an overall presentation of the OAEI 2012 campaign.

* This paper improves on the "Preliminary results" initially published in the on-site proceedings of the ISWC workshop on Ontology Matching (OM-2012). The only official results of the campaign, however, are on the OAEI web site.

1 Introduction

The Ontology Alignment Evaluation Initiative[1] (OAEI) is a coordinated international initiative, which organizes the evaluation of the increasing number of ontology matching systems [12; 10; 26]. The main goal of OAEI is to compare systems and algorithms on the same basis and to allow anyone for drawing conclusions about the best matching strategies. Our ambition is that, from such evaluations, tool developers can improve their systems.

Two first events were organized in 2004: (i) the Information Interpretation and Integration Conference (I3CON) held at the NIST Performance Metrics for Intelligent Systems (PerMIS) workshop and (ii) the Ontology Alignment Contest held at the Evaluation of Ontology-based Tools (EON) workshop of the annual International Semantic Web Conference (ISWC) [27]. Then, a unique OAEI campaign occurred in 2005 at the workshop on Integrating Ontologies held in conjunction with the International Conference on Knowledge Capture (K-Cap) [1]. Starting from 2006 through 2011 the OAEI campaigns were held at the Ontology Matching workshops collocated with ISWC [11; 9; 3; 6; 7; 8]. In 2012, the OAEI results will be presented again at the Ontology Matching workshop[2] collocated with ISWC, in Boston, USA.

Since last year, we have been promoting an environment for automatically processing evaluations (§2.2), which has been developed within the SEALS (Semantic Evaluation At Large Scale) project[3]. SEALS provided a software infrastructure, for automatically executing evaluations, and evaluation campaigns for typical semantic web tools, including ontology matching. An intermediate campaign was executed in March 2012 in coordination with the Second SEALS evaluation campaigns. This campaign, called OAEI 2011.5, had five tracks and 18 participants, and it only ran on the SEALS platform. The results of OAEI 2011.5 have been independently published on the OAEI web site and are now integrated in this paper with those of OAEI 2012. For OAEI 2012, almost all of the OAEI data sets were evaluated under the SEALS modality, providing a more uniform evaluation setting.

This paper synthetizes the 2012 evaluation campaign and introduces the results provided in the papers of the participants. The remainder of the paper is organized as follows. In Section 2, we present the overall evaluation methodology that has been used. Sections 3-9 discuss the settings and the results of each of the test cases. Section 10 overviews lessons learned from the campaign. Finally, Section 11 concludes the paper.

2 General methodology

We first present the test cases proposed this year to the OAEI participants (§2.1). Then, we discuss the resources used by participants to test their systems and the execution environment used for running the tools (§2.2). Next, we describe the steps of the OAEI campaign (§2.3-2.5) and report on the general execution of the campaign (§2.6).

[1] http://oaei.ontologymatching.org
[2] http://om2012.ontologymatching.org
[3] http://www.seals-project.eu

2.1 Tracks and test cases

This year's campaign consisted of 7 tracks gathering 9 data sets and different evaluation modalities:

The benchmark track (§3): Like in previous campaigns, a systematic benchmark series has been proposed. The goal of this benchmark series is to identify the areas in which each matching algorithm is strong or weak by systematically altering an ontology. This year, like in OAEI 2011, we used new systematically generated benchmarks, based on four ontologies other than the original bibliographic one. For three of these ontologies, the evaluation was performed in blind mode.

The expressive ontologies track offers real world ontologies using OWL modelling capabilities:

Anatomy (§4): The anatomy real world case is about matching the Adult Mouse Anatomy (2744 classes) and a part of the NCI Thesaurus (3304 classes) describing the human anatomy.

Conference (§5): The goal of the conference task is to find all correct correspondences within a collection of ontologies describing the domain of organizing conferences (the domain being well understandable for every researcher). Results were evaluated automatically against reference alignments and by using logical reasoning techniques.

Large biomedical ontologies (§8): This track aims at finding alignments between large and semantically rich biomedical ontologies such as FMA, SNOMED CT, and NCI. The UMLS Metathesaurus has been selected as the basis for the track's reference alignments.

Multilingual

Multifarm(§6): This dataset is composed of a subset of the Conference dataset, translated in eight different languages (Chinese, Czech, Dutch, French, German, Portuguese, Russian, and Spanish) and the corresponding alignments between these ontologies.

Directories and thesauri

Library(§7): The library track is a real-word task to match two thesauri. The goal of this track is to find whether the matchers can handle such lightweight ontologies including a huge amount of concepts and additional descriptions. Results are evaluated both against a reference alignment and through manual scrutiny.

Instance matching (§9): The goal of the instance matching track is to evaluate the performance of different tools on the task of matching RDF individuals which originate from different sources but describe the same real-world entity. Instance matching is organized in two sub-tasks:

Sandbox: The Sandbox is a simple dataset that has been specifically conceived to provide examples of some specific matching problems (like name spelling and other controlled variations). This is intended to serve as a test for those tools that are in an initial phase of their development process and/or for tools that are facing very focused tasks, such as person name matching.

IIMB: IIMB is an OWL-based dataset that is automatically generated by introducing a set of controlled transformations in an initial OWL Abox, in order: i) to provide an evaluation dataset for various kinds of data transformations, including value transformations, structural transformations, and logical transformations; ii) to cover a wide spectrum of possible techniques and tools.

Table 1 summarizes the variation in the tests under consideration.

test	formalism	relations	confidence	modalities	language	SEALS
benchmark	OWL	=	[0 1]	blind+open	EN	√
anatomy	OWL	=	[0 1]	open	EN	√
conference	OWL-DL	=, <=	[0 1]	blind+open	EN	√
multifarm	OWL	=	[0 1]	open	CZ, CN, DE, EN, ES, DE, FR, RU, PT	√
library	OWL	=	[0 1]	open	EN, DE	√
large bio	OWL	=	[0 1]	open	EN	√
sandbox	RDF	=	[0 1]	open	EN	
iimb	RDF	=	[0 1]	open	EN	

Table 1. Characteristics of the test cases (open evaluation is made with already published reference alignments and blind evaluation is made by organizers from reference alignments unknown to the participants).

We do not present the New York Times (NYT) sub-track, held in the instance matching track, since it had only one participant.

2.2 The SEALS platform

In 2010, participants of the Benchmark, Anatomy and Conference tracks were asked for the first time to use the SEALS evaluation services: they had to wrap their tools as web services and the tools were executed on the machines of the tool developers [28]. Since 2011, tool developers had to implement a simple interface and to wrap their tools in a predefined way including all required libraries and resources. A tutorial for tool wrapping was provided to the participants. This tutorial describes how to wrap a tool and how to use a simple client to run a full evaluation locally. After local tests are passed successfully, the wrapped tool was uploaded for a test on the SEALS portal[4]. Consequently, the evaluation was executed by the organizers with the help of the SEALS technology. This approach allowed to measure runtime and ensured the reproducibility of the results. As a side effect, this approach ensures also that a tool is executed with the same settings for all of the six tracks that were executed in the SEALS mode. This was already requested in the previous years, however, this rule was sometimes ignored by participants.

2.3 Preparatory phase

Ontologies to be matched and (where applicable) reference alignments have been provided in advance during the period between June 15^{th} and July 1^{st}, 2012. This gave

[4] `http://www.seals-project.eu/join-the-community/`

potential participants the occasion to send observations, bug corrections, remarks and other test cases to the organizers. The goal of this preparatory period is to ensure that the delivered tests make sense to the participants. The final test base was released on July 6^{th}, 2012. The data sets did not evolve after that.

2.4 Execution phase

During the execution phase, participants used their systems to automatically match the test case ontologies. In most cases, ontologies are described in OWL-DL and serialized in the RDF/XML format [4]. Participants can self-evaluate their results either by comparing their output with reference alignments or by using the SEALS client to compute precision and recall. They can tune their systems with respect to the non blind evaluation as long as the rules published on the OAEI web site are satisfied. This phase has been conducted between July 6^{th} and August 31^{st}, 2012.

2.5 Evaluation phase

Participants have been encouraged to provide (preliminary) results or to upload their wrapped tools on the SEALS portal by September 1^{st}, 2012. For the SEALS modality, a full-fledged test including all submitted tools has been conducted by the organizers and minor problems were reported to some tool developers, until finally a properly executable version of all the tools has been uploaded on the SEALS portal.

First results were available by September 22^{nd}, 2012. The track organizers provided these results individually to the participants. The results were published on the respective web pages by the track organizers by October 15^{th}. The standard evaluation measures are precision and recall computed against the reference alignments. For the matter of aggregation of the measures, we used weighted harmonic means (weights being the size of the true positives). Another technique that was used is the computation of precision/recall graphs so it was advised that participants provide their results with a weight to each correspondence they found. We also computed for some tracks the degree of alignment coherency. Additionally, we measured runtimes for all tracks conducted under the SEALS modality.

2.6 Comments on the execution

For a few years, the number of participating systems has remained roughly stable: 4 participants in 2004, 7 in 2005, 10 in 2006, 17 in 2007, 13 in 2008, 16 in 2009, 15 in 2010, 18 in 2011, 21 in 2012. However, participating systems are now constantly changing. In 2012, 7 systems have not participated in any of the previous campaigns. The list of participants is summarized in Table 2.

This year only four systems participated in the instance matching track; two of them (LogMap and LogMapLt) participated also in the SEALS tracks.

Two tools, OMR and OntoK, are not shown in the table and were not included in the final evaluation.

OMR generated alignments with correspondences containing non-existing entities in one or both of the ontologies being matched. Moreover, it seems that its alignments

System	Aroma	ASE	AUTOMSv2	CODI	GOMMA	Hertuda	Hotmatch	LogMap	LogMapLt	MaasMtch	MapSSS	MEDLEY	Optima	SBUEI	semsim	ServOMap	ServOMapLt	TOAST	WeSeE	WikiMatch	YAM++	Total=21
Confidence	√	√	√		√	√		√		√		√				√	√	√	√	√	√	14
benchmarks	√	√	√		√	√	√	√	√	√	√	√	√			√	√		√	√	√	17
anatomy	√			√	√	√	√	√	√	√	√		√			√	√	√	√	√	√	16
conference	√	√	√	√	√	√	√	√	√	√	√	√	√			√	√		√	√	√	18
multifarm	√	√	√	√	√	√	√	√	√	√	√	√	√			√	√		√	√	√	18
library	√			√	√	√	√	√	√		√		√			√	√		√		√	13
large bio	√		√		√	√	√	√	√	√	√					√	√			√	√	13
sandbox								√	√					√								3
iimb								√	√					√	√							4
total	6	3	4	4	6	6	6	8	8	5	6	3	5	2	1	6	6	1	5	5	6	102

Table 2. Participants and the state of their submissions. Confidence stands for the type of result returned by a system: it is ticked when the confidence is a non boolean value.

are iteratively composed one by one, and that the last alignment contains all correspondences from other alignments but with changed namespaces, leading in many cases to very low precisions with quite high recalls. This behavior was reproduced along all the tracks.

OntoK revealed several bugs when preliminary tests were executed; the developers were not able to fix all of them before proceeding to the final evaluation.

Another tool was disqualified because we found that in quite a large number of cases across different tracks, the results it provided were too often exactly those of another matcher, including syntactic errors. After further scrutiny, it became obvious that the system implemented specific tricks to run well the OAEI tests instead of being a genuine matcher.

Finally, some systems were not able to pass some test cases as indicated in Table 2

The summary of the results track by track is presented in the following sections.

3 Benchmark

The goal of the benchmark data set is to provide a stable and detailed picture of each algorithm. For that purpose, algorithms are run on systematically generated test cases.

3.1 Test data

The systematic benchmark test set is built around a seed ontology and many variations of it. Variations are artificially generated, and focus on the characterization of the behavior of the tools rather than having them compete on real-life problems. They are organized in three groups:

Simple tests (1xx) such as comparing the reference ontology with itself;

Systematic tests (2xx) obtained by discarding/modifying features from the reference ontology. Considered features are names of entities, comments, the specialization hierarchy, instances, properties and classes.

Real-life ontologies (3xx) found on the web.

Full description of the systematic benchmark test set can be found on the OAEI web site.

This year the focus was on scalability, i.e., the ability of matchers to deal with data sets of increasing number of elements. To that extent, we departed from the usual bibliographic benchmark that has been used since 2004. We used a test generator [25] in order to reproduce the structure of benchmark for different seed ontologies, from different domains and with different sizes. We have generated five different benchmarks against which matchers have been evaluated:

benchmark (biblio) allows for comparison with other systems since 2004. The seed ontology concerns bibliographic references and is inspired freely from BibTeX. It contains 33 named classes, 24 object properties, 40 data properties, 56 named individuals and 20 anonymous individuals. This year we have used a new automatically generated version for this benchmark.

benchmark2 is related with the commerce domain. Its seed ontology contains 74 classes, 106 object properties and 35 named individuals.

benchmark3 is related with bioinformatics. Its seed ontology contains 233 classes, 83 object properties, 38 data properties and 681 named individuals.

benchmark4 is related with the product design domain. Its seed ontology contains 182 classes, 88 object properties, 202 data properties and 376 named individuals.

benchmark5 (finance) is based on the Finance ontology[5], which contains 322 classes, 247 object properties, 64 data properties and 1113 named individuals. It has been already considered in OAEI 2011 and 2011.5.

Having these five data sets also allowed us to better evaluate the dependency between the results and the seed ontology. **biblio** and **finance** were disclosed to the participants; the other benchmarks were tested in blind mode.

For all data sets, the reference alignments are still limited: they only match named classes and properties and use the "=" relation with confidence of 1.

3.2 Results

We evaluated 18 systems from the 22 participating in the SEALS tracks (Table 2). Besides OMR, OntoK and TOAST, excluded for reasons already explained, requirements for executing CODI in our machines were not met due to software license problems. In the following, we present the evaluation results.

[5] `http://www.fadyart.com/ontologies/data/Finance.owl`

Compliance Benchmark compliance tests have been executed on two cores and 8GB RAM Debian virtual machines (VM) running continuously in parallel, except for the finance data set which required 10GB RAM for some systems. For each benchmark seed ontology, data sets of 94 tests were automatically generated. We excluded from the whole systematic benchmark test set (111 tests), the tests that were not automatically generated: 102–104, 203–210, 230–231, 301–304.

Table 3 shows the compliance results (harmonic means of precision, F-measure and recall) of the five benchmark data sets for all the participants, as well as those given by edna, a simple edit distance algorithm on labels which is used as a baseline. The table also presents the confidence-weighted values of the same parameters.

Only ASE presented problems to process the **finance** data set, and MEDLEY did not completed the evaluation of the **benchmark4** and the **finance** data sets in a reasonable amount of time (12 hours).

Table 3 shows that, with few exceptions, all systems achieve higher levels of precision than recall for all benchmarks. Besides, no tool had a worst precision performance than the baseline, and only ServOMapLt had a significantly lower recall, with LogMap having slightly lower values for the same measure.

For those systems which have provided their results with confidence measures different from 1 or 0 (see Table 2), it is possible to draw precision/recall graphs and to compute weighted precision and recall. Systems providing accurate confidence values are rewarded by these measures [7]. Precision is increased for systems with many incorrect correspondences and low confidence, like edna and MaasMatch. Recall is decreased for systems with apparently many correct correspondences and low confidence, like AROMA, LogMap and YAM++. The variation for YAM++ is quite impressive, especially for the **biblio** benchmark.

Precision/recall graphs are given in Figure 1. The graphs show the real precision at n% recall and they stop when no more correspondences are available; then the end point corresponds to the precision and recall reported in Table 3.

Comparison across data sets From the results in Table 3, we observe that on average, all matchers have better performance than the baseline. The group of best systems in each data set remains relatively the same across the different benchmarks: YAM++, MapSSS and AROMA seems to generate the best alignments in terms of F-measure.

We also observe a high variance in the results of some systems across different benchmarks. Outliers are, for example, a poor precision for AROMA with **benchmark3** and a poor recall for ServOMapLt with **biblio**. These variations suggest interdependencies between matching systems and datasets that would need additional analysis requiring a deep knowledge of the evaluated systems. Such information is, in particular, useful for developers to detect and fix problems specific to their tools.

We also compare the results obtained by the tools that have participated in OAEI 2011 and 2012 on **biblio** and **finance** benchmarks which have been used in both campaigns. With respect to **biblio**, we observe negative variations between 2-4% for some tools, as well as positive variations between 1-3% for others. Regarding **finance**, the number of systems able to pass the tests increased, and for many tools that passed the tests in previous campaigns, positive variations between 1-3% were observed.

system	**refalign**			**edna**			**AROMA**			**ASE**			**AUTOMSv2**			**GOMMA**			**Hertuda**		
test	P	F	R	P	F	R	P	F	R	P	F	R	P	F	R	P	F	R	P	F	R
biblio	1.0	1.0	1.0	.35(.45)	.41(.47)	.50	.98(.99)	.77(.73)	.64(.58)	.49	.51(.52)	.54	.97	.69	.54	.75(.74)	.67(.65)	.61(.58)	.90	.68	.54
benchmark2	1.0	1.0	1.0	.46(.61)	.48(.55)	.50	.97(.98)	.76(.73)	.63(.58)	.72(.74)	.61	.53	.97	.68	.52	.97	.69(.67)	.53(.51)	.93	.67	.53
benchmark3	1.0	1.0	1.0	.22(.25)	.30(.33)	.50	.38(.43)	.53(.54)	.83(.73)	.27	.36	.54	.99(1.0)	.70	.54	.99(1.0)	.70(.69)	.54(.52)	.94	.68	.54
benchmark4	1.0	1.0	1.0	.31(.37)	.38(.42)	.50	.96	.73(.70)	.59(.55)	.40(.41)	.45	.51	.91(.92)	.65	.51(.50)	.86	.63(.62)	.50(.49)	.90	.66	.51
finance	1.0	1.0	1.0	.22(.25)	.30(.33)	.50	.94	.72(.70)	.58(.56)	n.a.	n.a.	n.a.	.35	.42(.39)	.55(.46)	.95(.94)	.66(.64)	.51(.49)	.72	.62	.55

	HotMatch			**LogMap**			**LogMapLt**			**MaasMatch**			**MapSSS**			**MEDLEY**			
test	P	F	R	P	F	R	P	F	R	P	F	R	P	F	R	P	F	R	
biblio	.96	.66	.50	.73	.56(.51)	.45(.39)	.71	.59	.50	.54(.90)	.56(.63)	.57(.49)	.99	.87	.77	.60(.59)	.54(.53)	.50 (.48)	
benchmark2	.99	.68	.52	1.0	.64(.59)	.47(.42)	.95	.66	.50	.60(.93)	.60(.65)	.60(.50)	1.0	.86	.76(.75)	.92(.94)	.65(.63)	.50 (.48)	
benchmark3	.99	.68	.52	.95(.96)	.65(.60)	.49(.44)	.95	.65	.50	.53(.90)	.53(.63)	.53(.48)	1.0	.82	.70	.78	.61(.56)	.50 (.43)	
benchmark4	.99	.66	.50	.99(1.0)	.63(.58)	.46(.41)	.95	.65	.50	.54(.92)	.54(.64)	.54(.49)	1.0(.99)	.81	.68	to	to	to	
finance	.97	.67	.51	.95(.94)	.63(.57)	.47(.40)	.90	.66	.52	.59(.92)	.59(.62)	.60(.48)	.99	.83	.71	to	to	to	

	Optima			**ServOMap**			**ServOMapLt**			**WeSeE**			**WikiMatch**			**YAM++**			
test	P	F	R	P	F	R	P	F	R	P	F	R	P	F	R	P	F	R	
biblio	.89	.63	.49	.88	.58	.43	1.0	.33	.20	.99	.69(.68)	.53(.52)	.74	.62	.54	.98(.95)	.83(.18)	.72(.10)	
benchmark2	1.0	.66	.50	1.0	.67	.50	1.0	.51	.35(.34)	1.0	.69(.68)	.52	.97	.67(.68)	.52	.96(1.0)	.89(.72)	.82(.56)	
benchmark3	.97	.69	.53	1.0	.67	.50	1.0	.55	.38	1.0	.70(.69)	.53	.96(.97)	.68	.52	.97(1.0)	.85(.70)	.76(.54)	
benchmark4	.92	.60	.45	.89	.60(.59)	.45(.44)	1.0	.41	.26	1.0	.67(.66)	.50	.94(.95)	.66	.51	.96(1.0)	.83(.70)	.72(.54)	
finance	.96	.66	.51	.92	.63	.48	.99	.51(.50)	.34	.99	.70(.69)	.54(.53)	.74(.75)	.62(.63)	.54	.97(1.0)	.90(.72)	.84(.57)	

Table 3. Results obtained by participants on the five benchmark test cases aggregated with harmonic means (values within parentheses are *weighted* version of the measure reported only when different).

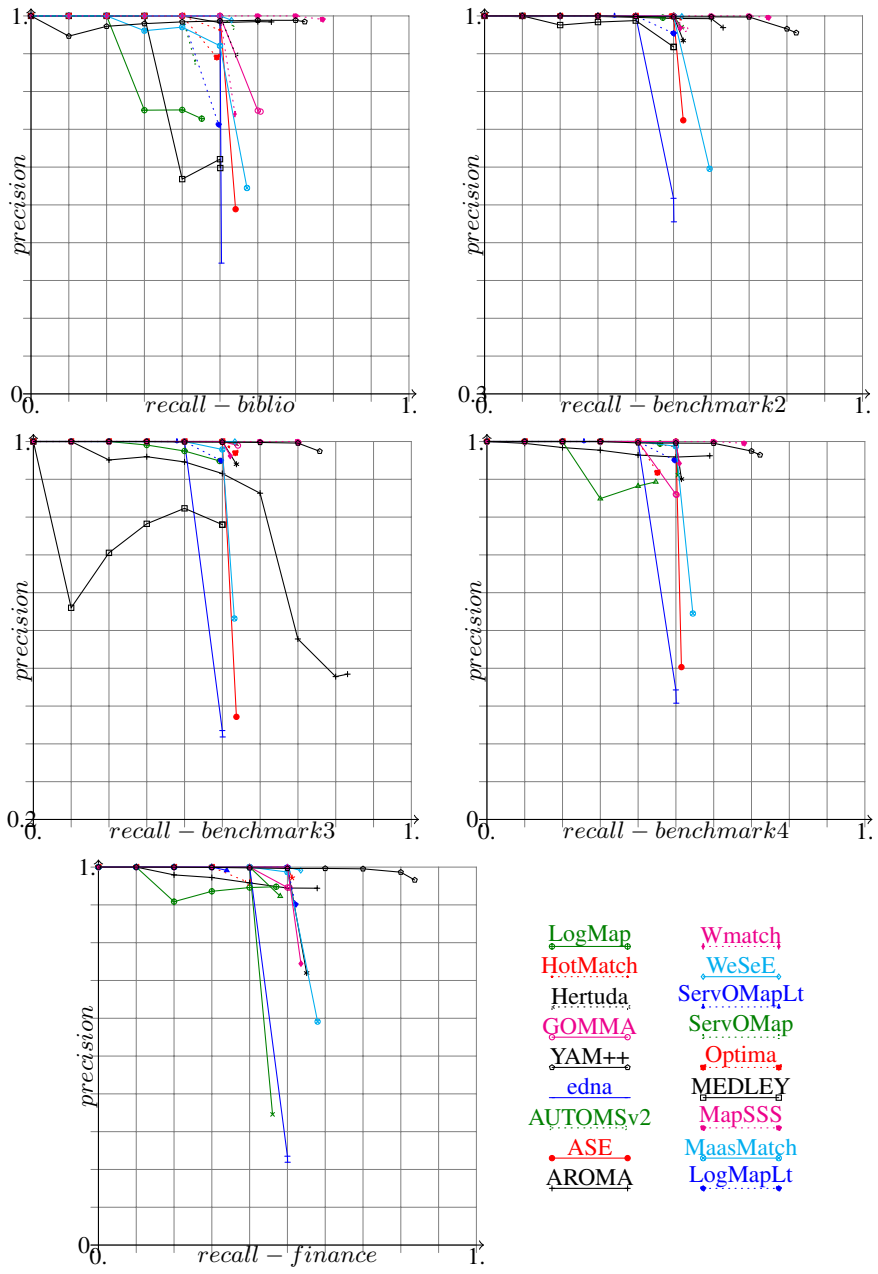

Fig. 1. Precision/recall graphs for benchmarks. The alignments generated by matchers are cut under a threshold necessary for achieving $n\%$ recall and the corresponding precision is computed. Systems for which these graphs are not meaningful (because they did not provide graded confidence values) are drawn in dashed lines.

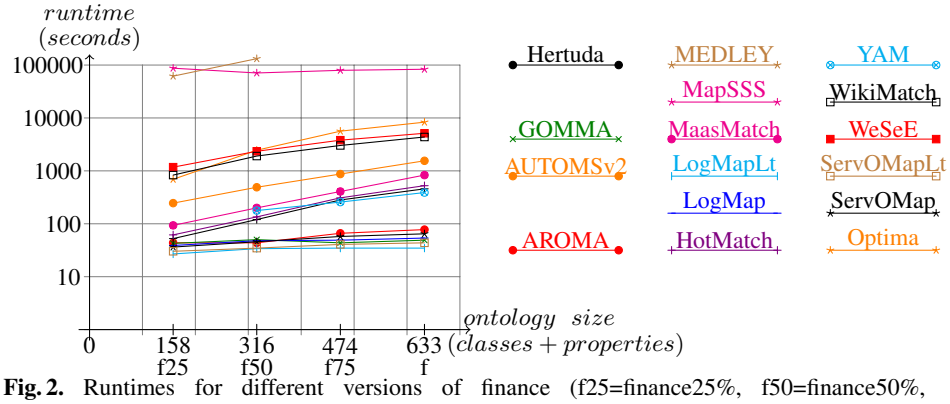

Fig. 2. Runtimes for different versions of finance (f25=finance25%, f50=finance50%, f75=finance75%, f=finance).

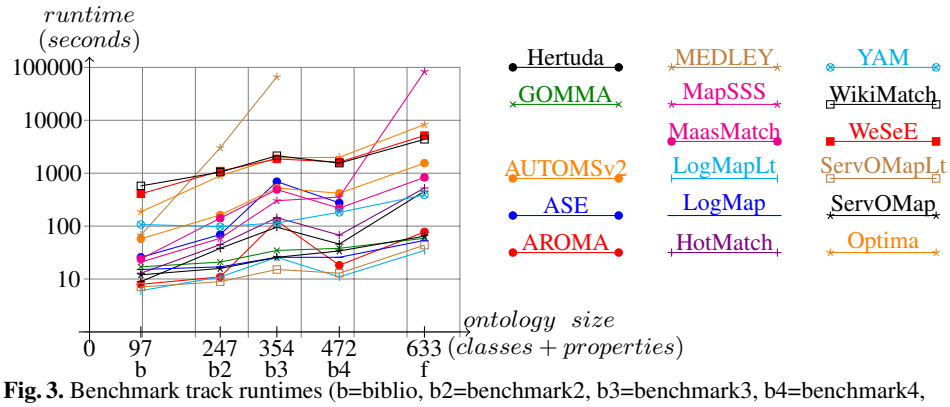

Fig. 3. Benchmark track runtimes (b=biblio, b2=benchmark2, b3=benchmark3, b4=benchmark4, f=finance).

Runtime Regarding runtime, scalability has been evaluated from two perspectives: on the one hand we considered the five seed ontologies from different domains and with different sizes; on the other hand we considered the **finance** ontology scaling it by reducing its size by different factors (25%, 50% and 75%). For the two modalities, the data sets were composed of a subset containing 15 tests extracted from a whole systematic benchmark.

All the experiments were done on a 3GHz Xeon 5472 (4 cores) machine running Linux Fedora 8 with 8GB RAM. Figures 2 and 3 show semi-log graphs for runtime measurements against data set sizes in terms of classes and properties.

First of all we observe that for the **finance** tests, the majority of tools have a monotonic increasing run time, with the exception of MapSSS which exhibits an almost constant response time. On the contrary, this does not happen for the benchmark tests, for which the **benchmark3** test causes a break in the monotonic behavior. One reason for this could be that the **benchmark3** ontology has a more complex structure than the other ones, and that matchers basing their work in structural analysis are more affected than others.

Figures also show that there is a set of tools that distance themselves from the others: LogMapLt, ServOMapLt, LogMap, ServOMap, GOMMA and Aroma are the fastest tools, and are able to process large ontologies in a short time. On the contrary, there exist tools that were not able to deal with large ontologies in the same conditions: MEDLEY and MapSSS fall in this category.

3.3 Conclusions

Having five different benchmarks allowed us to see the degree of dependency on the shapes and sizes of the seed ontologies. Even if differences were observed in the results obtained, we can conclude that excepting a few cases, the tools do not show large variations in the results to the different benchmarks.

Regarding compliance, we observed that with very few exceptions, the systems performed always better than the baseline. However, there were no significant improvements in the performance of the systems with respect to their performance in last OAEI campaigns (OAEI 2011 and 2011.5).

Regarding runtime, we noticed that given ontologies of different sizes sharing the structure and knowledge domain to a big extent, the response time follows generally the shape of a monotonic increasing function. On the contrary, this is not always true if the shapes or the knowledge domains of the ontologies change.

The results obtained this year allow us to confirm that we cannot conclude on a general correlation between runtime and quality of alignments. The slowest tools do not necessarily provide the best compliance results.

4 Anatomy

The anatomy track confronts matchers with a specific type of ontologies from the biomedical domain. In this domain, many ontologies have been built covering different aspects of medical research. We focus on two fragments of biomedical ontologies which describe the human anatomy and the anatomy of the mouse. The data set of this track has been used since 2007 with some improvements over the last years. For a detailed description, we refer the reader to the OAEI 2007 results paper [9].

4.1 Experimental setting

Contrary to previous years, we conducted only a single evaluation experiment by executing each matcher in its standard setting. In our experiments, we compare precision, recall, F-measure and recall+. The measure recall+ indicates the amount of detected non-trivial correspondences. The matched entities in a non-trivial correspondence do not have the same normalized label. The approach that generates only trivial correspondences is depicted as baseline *StringEquiv* in the following section. In OAEI 2011/2011.5, we executed the systems on our own (instead of analyzing submitted alignments) and reported about measured runtimes. Unfortunately, we did not use exactly the same machine compared to previous years. Thus, runtime results are not fully comparable across years. In 2012, we used an Ubuntu machine with 2.4 GHz (2

cores) and 3GB RAM allocated to the matching systems. Further, we used the SEALS client to execute our evaluation. However, we slightly changed the way precision and recall are computed, i.e., the results generated by the SEALS client vary in some cases by 0.5% compared to the results presented below. In particular, we remove trivial correspondences in the `oboInOwl` namespace like

```
http://...oboInOwl#Synonym = http://...oboInOwl#Synonym
```

as well as correspondences expressing relations different from equivalence. We also checked whether the generated alignment is coherent, i.e., there are no unsatisfiable concepts when the ontologies are merged with the alignment.

4.2 Results

In Table 4, we listed all the participating systems that generated an alignment in less than ten hours. The listing comprises 17 entries. Three systems participated each with two different versions. This is GOMMA (the extension "'-bk"' refers to the usage of background knowledge), LogMap and ServoMap (both systems have submitted an additional lightweight version that uses only some core components). Thus, 14 different systems generated an alignment within the given time frame. There were three participants ASE, AUTOMSv2, and MEDLEY that did no finish in time or threw an exception. Due to several hardware and software requirements, we could not install TOAST on the machine on which we executed the other systems. We executed the matcher on a different machine of similar strength. For this reason, the runtime of TOAST is not fully comparable to the other runtimes (indicated by an asterisk).

Compared to previous years, we can observe a clear speed increase. In 2012, five systems (counting two versions of the same system as one) finished in less than 100 seconds, compared to two systems in OAEI 2011 and three systems in OAEI 2011.5. This has to be mentioned as a positive trend. Moreover, in 2012 we were finally able to generate results for 14 of 17 systems, while in 2011 only 7 of 14 systems generated results of acceptable quality within the given time frame. The top systems in terms of runtimes are GOMMA, LogMap and ServOMap. Depending on the specific version of the systems, they require between 6 and 34 seconds to match the ontologies. Table 4 shows that there is no correlation between the quality of the generated alignment in terms of precision and recall and the required runtime. This result has also been observed in previous campaigns.

Table 4 also shows the results for precision, recall and F-measure. We ordered the matching systems with respect to the achieved F-measure. The F-measure is an aggregation of precision and recall. Depending on the application for which the generated alignment is used, it might, for example, be more important to favor precision over recall or vice versa. In terms of F-measure, GOMMA-bk is ahead of the other participants. The differences of GOMMA-bk compared to GOMMA (and the other systems) are based on mapping composition techniques and the reuse of mappings between UMLS, Uberon and FMA. GOMMA-bk is followed by a group of matching systems (YAM++, CODI, LogMap, GOMMA) generating alignments that are very similar with respect to precision, recall and F-measure (between 0.87 and 0.9 F-measure). To our knowledge,

Matcher	Runtime(s)	Size	Precision	F-measure	Recall	Recall+	Coherent
GOMMA-bk	15	1534	0.917	0.923	0.928	0.813	-
YAM++	69	1378	0.943	0.898	0.858	0.635	-
CODI	880	1297	0.966	0.891	0.827	0.562	√
LogMap	20	1392	0.920	0.881	0.845	0.593	√
GOMMA	17	1264	0.956	0.870	0.797	0.471	-
MapSSS	453	1212	0.935	0.831	0.747	0.337	-
WeSeE	15833	1266	0.911	0.829	0.761	0.379	-
LogMapLt	6	1147	0.963	0.829	0.728	0.290	-
TOAST	3464*	1339	0.854	0.801	0.755	0.401	-
ServOMap	34	972	0.996	0.778	0.639	0.054	-
ServOMapL	23	976	0.990	0.775	0.637	0.052	-
HotMatch	672	989	0.979	0.773	0.639	0.145	-
AROMA	29	1205	0.865	0.766	0.687	0.321	-
StringEquiv	-	946	0.997	0.766	0.622	0.000	-
Wmatch	17130	1184	0.864	0.758	0.675	0.157	-
Optima	6460	1038	0.854	0.694	0.584	0.133	-
Hertuda	317	1479	0.690	0.681	0.673	0.154	-
MaasMatch	28890	2737	0.434	0.559	0.784	0.501	-

Table 4. Comparison against the reference alignment, runtime is measured in seconds, the "size" column refers to the number of correspondences in the generated alignment.

these systems either do not use specific background knowledge for the biomedical domain or use it only in a very limited way. The results of these systems are at least as good as the results of the best system in OAEI 2007-2010, only AgreementMaker, using additional background knowledge, could generate better results in OAEI 2011. Most of the evaluated systems achieve an F-measure that is higher than the baseline that is based on (normalized) string equivalence. Moreover, nearly all systems find many non-trivial correspondences. An exception is the system ServOMap (and its lightweight version) that generates an alignment that is quite similar to the alignment generated by the baseline approach.

Concerning alignment coherency, only CODI and LogMap generated coherent alignments. We have to conclude that there have been no improvements compared to OAEI 2011 with respect to taking alignment coherence into account. LogMap and CODI generated a coherent alignment already in 2011. Furthermore, is can be observed (see Section 5) that YAM++ generates coherent alignments for the ontologies of the Conference track, which are much smaller but more expressive, while it fails to generate coherent alignments for larger biomedical ontologies (see also Section 8). This might be based on using different settings for larger ontologies to avoid reasoning problems with larger input.

4.3 Conclusions

Most of the systems top the string equivalence baseline with respect to F-measure. Moreover, we reported that several systems achieve very good results compared to the

evaluations of the previous years. A clear improvement compared to previous years can be seen in the number of systems that are able to generate such results. It is also a positive trend that more matching systems can create good results within short runtimes. This might partially be caused by offering the Anatomy track constantly in its current form over the last six years together with publishing matcher runtimes. At the same time, new tracks that deal with large (and very large) matching tasks are offered. These tasks can only be solved with efficient matching strategies that have been implemented over the last years.

5 Conference

The conference test case introduces matching several moderately expressive ontologies. Within this track, participant results were evaluated against reference alignments (containing merely equivalence correspondences) and by using logical reasoning. As last year, the evaluation has been supported by the SEALS technology. This year we used refined and harmonized reference alignments.

5.1 Test data

The collection consists of sixteen ontologies in the domain of organizing conferences. These ontologies have been developed within the OntoFarm project[6].

The main features of this test case are:

- *Generally understandable domain.* Most ontology engineers are familiar with organizing conferences. Therefore, they can create their own ontologies as well as evaluate the alignments among their concepts with enough erudition.
- *Independence of ontologies.* Ontologies were developed independently and based on different resources, they thus capture the issues in organizing conferences from different points of view and with different terminologies.
- *Relative richness in axioms.* Most ontologies were equipped with OWL DL axioms of various kinds; this opens a way to use semantic matchers.

Ontologies differ in their numbers of classes, of properties, in expressivity, but also in underlying resources.

5.2 Results

This year, we provide results in terms of $F_{0.5}$-measure, F_1-measure and F_2-measure, comparison with baseline matcher, precision/recall triangular graph and coherency evaluation.

[6] http://nb.vse.cz/~svatek/ontofarm.html

Evaluation based on reference alignments We evaluated the results of participants against new reference alignments (labelled as *ra2* on the conference web-page). This includes all pairwise combinations between 7 different ontologies, i.e. 21 alignments.

New reference alignments have been generated as a transitive closure computed on the original reference alignments. In order to obtain a coherent result, conflicting correspondences, i.e., those causing unsatisfiability, have been manually inspected and removed. As a result the degree of correctness and completeness of the new reference alignment is probably slightly better than for the old one. However, the differences are relatively limited. Whereas the new reference alignments are not open, the old reference alignments (labeled as *ra1* on the conference web-page) are available. These represent close approximation of the new ones.

Matcher	Precision	$F_{0.5}$-measure	F_1-measure	F_2-measure	Recall
YAM++	.78	.75	.71	.67	.65
LogMap	.77	.71	.63	.57	.53
CODI	.74	.69	.63	.58	.55
Optima	.60	.61	.61	.62	.63
GOMMA	.79	.68	.56	.47	.43
Hertuda	.70	.63	.56	.49	.46
MaasMatch	.60	.58	.56	.53	.52
Wmatch	.70	.63	.55	.48	.45
WeSeE	.72	.64	.55	.48	.44
HotMatch	.67	.62	.55	.50	.47
LogMapLt	.68	.62	.54	.48	.45
Baseline	*.76*	*.64*	*.52*	*.43*	*.39*
ServOMap	.68	.60	.51	.45	.41
ServOMapLt	.82	.65	.50	.41	.36
MEDLEY	.59	.55	.49	.45	.42
ASE*	.61	.55	.48	.43	.40
MapSSS	.47	.47	.46	.46	.46
AUTOMSv2*	.64	.54	.44	.37	.33
AROMA	.33	.34	.37	.39	.41

Table 5. The highest average $F_{[0.5|1|2]}$-measure and their corresponding precision and recall for each matcher F_1-optimal threshold.

Table 5 shows the results of all participants with regard to the new reference alignment. $F_{0.5}$-measure, F_1-measure and F_2-measure are computed for the threshold that provides the highest average F_1-measure. F_1 is the harmonic mean of precision and recall where both are equally weighted; F_2 weights recall higher than precision and $F_{0.5}$ weights precision higher than recall. The matchers shown in the table are ordered according to their highest average F_1-measure. Our *baseline*[7] divides matchers into two groups. Group 1 consists of matchers (YAM++, LogMap, CODI, Optima, GOMMA, Hertuda, MaasMatch, Wmatch, WeSeE, HotMatch and LogMapLt) having better (or equal) results than *Baseline*. Other matchers (ServOMap, ServOMapLt, MEDLEY,

[7] String matcher based on string equality applied on local names of entities which were lowercased before.

ASE, MapSSS, AUTOMSv2 and AROMA) performed worse than *baseline*. There are two matchers (ASE and AUTOMSv2) with asterisks which did not generate 3 out of 21 alignments. Thus, their results are just an approximation.

Performance of matchers from Group 1 regarding F_1-measure is visualized in Figure 4.

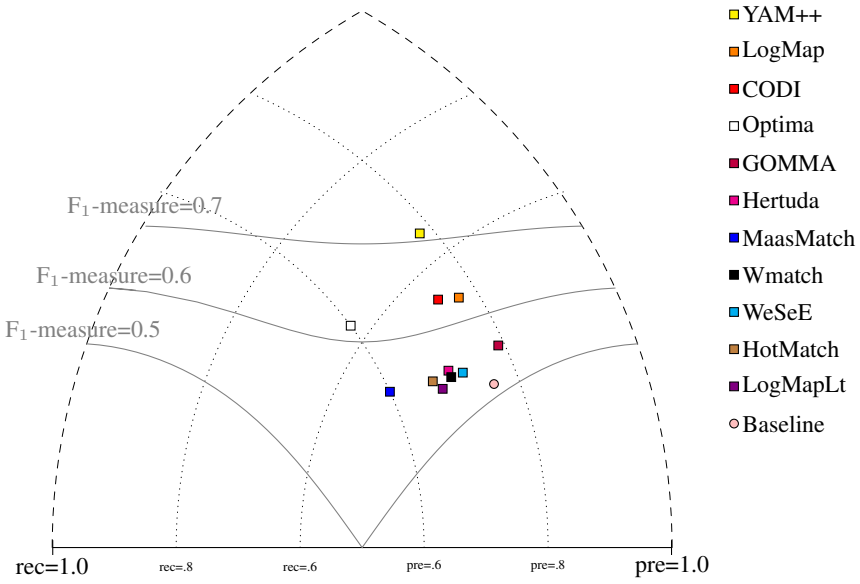

Fig. 4. Precision/recall triangular graph for the conference track. Matchers are represented as squares and Baseline is represented as a circle. Dotted lines depict level of precision/recall while values of F_1-measure are depicted by areas bordered by corresponding lines F_1-measure=0.[5|6|7].

Comparison with previous years Seven matchers also participated in OAEI 2011 and 10 matchers participated in OAEI 2011.5. The largest improvement was achieved by Optima (precision from .23 to .60 and recall from .52 to .63) and YAM++ (precision from .74 to .78 and recall from .51 to .65) between OAEI 2011 and 2012. Four matchers were improved between OAEI 2011.5 and 2012 and five matchers were improved between OAEI 2011 and 2012.

Runtimes We measured the total time of generating all 120 alignments. It was executed on a laptop with Ubuntu machine running on Intel Core i5, 2.67GHz and 4GB RAM. In all, four matchers finished all 120 test cases within 1 minute (LogMapLt - 44 seconds, Hertuda - 49 seconds, ServOMapLt - 50 seconds and AROMA - 55 seconds). Next, four matchers needed less than 2 minutes (ServOMap, HotMatch, GOMMA and ASE). 10 minutes were enough for the next four matchers (LogMap, MapSSS, MaasMatch and AUTOMSv2). Finally, 5 matchers needed up to 40 minutes to finish all 120 test cases (Optima - 22 min, MEDLEY - 30 min, WeSeE - 36 min, CODI - 39 min and Wmatch - 40 min). YAM++ did not finish the task of matching all 120 test cases within five hours.

In conclusion, regarding performance we can see (clearly from Figure 4) that YAM++ is on the top. Next three matchers (LogMap, CODI, and Optima) are relatively close to each other. This year there is a largest group of matchers which are above *baseline* than previous years. Moreover, it is very positive that several matchers managed to improve their performance in such a short time as one year or even half a year.

Evaluation based on alignment coherence As in previous years, we applied the Maximum Cardinality measure to evaluate the degree of alignment incoherence. Details on this measure and its implementation can be found in [19]. The results of our experiments are depicted in Table 6. Contrary to last year, we only compute the average for all test cases of the conference track for which there exists a reference alignment. The presented results are thus aggregated mean values for 21 test cases. In some cases we could not compute the degree of incoherence due to the combinatorial complexity of the problem. In this case we were still able to compute a lower bound for which we know that the actual degree is (probably only slightly) higher. Such results are marked with a *. Note that we only included in our evaluation those matchers that generated alignments for all test cases of the subset with reference alignments.

Matcher	AROMA	CODI	GOMMA	Hertuda	HotMatch	LogMap	LogMapLt	MaasMatch*
Alignment Size	20.9	11.1	8.2	9.9	10.5	10.3	9.9	83.1
Incoherence Degree	19.4%	0%	1.1%	5.2%	5%	0%	5.4%	24.3%
Incoherent Alignments	18	0	2	9	9	0	7	20

Matcher	MapSSS	MEDLEY*	Optima	ServOMap	ServOMapLt	WeSeE	Wmatch	YAM++
Alignment Size	14.8	55.6	15.9	9	6.5	9.4	9.9	12.5
Incoherence Degree	12.6%	30.5%	7.6%	3.3%	0%	3.2%	6%	0%
Incoherent Alignments	18	20	12	5	0	6	10	0

Table 6. Average size of alignments, average degree of incoherence, and number of incoherent alignments. The mark * is added if we only provide lower bound of the degree of incoherence due to the combinatorial complexity of the problem.

Four matchers can generate coherent alignments. These matchers are CODI, LogMap, ServOMapLt, and YAM++. However, it is not always clear whether this is related to a specific approach that tries to ensure the coherency, or whether this is only indirectly caused by generating small and highly precise alignments. In particular, the coherence of the alignments from ServOMapLt, which does not apply any semantic technique, might be caused by such an approach. The matcher generates overall the smallest alignments. Because there are some matchers that cannot generate a coherent alignment for alignments that have in average a size from 8 to 12 correspondences, it can

be assumed that CODI, LogMap, and YAM++ have implemented specific coherency-preserving methods. Those matchers generate also between 8 to 12 correspondences, however, none of their alignments is incoherent. This is an important improvement compared to the previous years, for which we observed that only one or two matchers managed to generate (nearly) coherent alignments.

6 MultiFarm

In order to be able to evaluate the ability of matching systems to deal with ontologies in different languages, the MultiFarm dataset has been proposed [21]. This dataset results from the translation of seven Conference track ontologies (cmt, conference, confOf, iasted, sigkdd, ekaw and edas), in eight languages: Chinese, Czech, Dutch, French, German, Portuguese, Russian, and Spanish (+ English). The translations in 8 languages + English result in 36 pairs of languages. Overall, we have 36×49 matching tasks (see [21] for details on the pairs of languages and ontologies).

6.1 Experimental setting

For the 2012 evaluation campaign, we have used a subset of the whole MultiFarm dataset, omitting all the pairs of matching tasks involving the ontologies edas and ekaw (resulting in 36×25 matching tasks). This allows for using the omitted test cases as blind evaluation tests in the future. Contrary to OAEI 2011.5, we have included the Chinese and Russian translations.

Within the MultiFarm dataset, we can distinguish two types of matching tasks: (i) those test cases in which two different ontologies have been translated in different languages (cmt–confOf, for instance); and (ii) those test cases where the same ontology has been translated in different languages (cmt–cmt, for instance). For the test cases of type (ii), good results are not directly related to the use of specific techniques for dealing with ontologies in different natural languages, but on the ability to exploit the fact that both ontologies have an identical structure (and that the reference alignment covers all entities described in the ontologies).

This year, seven participating systems (out of 21 systems participated in OAEI, see Table 2) use specific multilingual methods: ASE, AUTOMSv2, GOMMA, MEDLEY, WeSeE, Wmatch, and YAM++. The other systems are not specifically designed to match ontologies in different languages, nor do they make use of a component that can be used for that purpose.

ASE (a version of AUTOMSv2) uses the Microsoft Bing Translator API for translating the ontologies to English. This process is performed before ASE profiling, configuration and matching methods are executed, so its input will consider only English labeled copies of ontologies. AUTOMSv2 follows a similar approach, but re-using a free Java API named WebTranslator. GOMMA uses a free translation API (MyMemory), for translating non-English concept labels to English. The translations are associated to concepts as new synonyms. Iteratively, GOMMA creates a bilingual dictionary for each ontology, which is used within the matching process. WeSeE and YAM++, as AUTOMS2, use Microsoft Bing Translation for translating the labels contained in

the input ontologies to English. Then, the translated English ontologies are matched using standard matching procedures of WeSeE and YAM++. Finally, Wmatch exploits Wikipedia for extracting inter-language. All matchers (with the exception of Wmatch) use English as a pivot language. MEDLEY is the only matcher for which we have no information on the techniques it exploits to deal with multilingualism.

6.2 Execution setting and runtime

All systems (with the exception of CODI) have been executed on a 3GHz Xeon 5472 (4 cores) machine, running Linux Fedora 8 with 8GB RAM. The runtimes for each system can be found in Table 7. As CODI has been executed on a different setting, its runtime cannot be compared with the runtime of other systems. We observe large differences between the time required for a system to complete the 36×25 matching tasks. While WeSeE requires $\cong$ 15 minutes, Wmatch takes $\cong$ 17 hours. It is mainly due to the fact that Wmatch uses an external resource (Wikipedia) for looking for inter-languages links. This requires considerably more time than simpler requests for translations.

6.3 Overall results

Before discussing the results per pairs of languages, we present the aggregated results for the test cases within type (i) and (ii) matching task. Table 7 shows the aggregated results. Systems not listed in this table have generated empty alignments, for all test cases (ServOMap and ServOMapL), have thrown exceptions (ASE, OMR, OntoK), or have not been evaluated due to their execution requirements (TOAST). AROMA was not able to generate alignments for test cases of type (i).

As shown in Table 7, we can observe significant differences between the results obtained for each type of matching task, specially in terms of precision. While the systems that implement specific multilingual techniques clearly generate the best results for test cases of type (i), only one of these systems (YAM++) is among the top (3) F-measures for type (ii) test cases. For these test cases, MapSSS and CODI, which implement strategies to deal with ontologies that share structural similarities, have better results. Due to this feature, they have preserved their overall performance this year (using the same version as for the last campaign), even though harder tests have been included in 2012 (Chinese and Russian translations). On the other hand, for the other matchers in the same situation, the differences in the results are explained by the presence of such harder tests cases this year.

Furthermore, as observed in the OAEI 2011.5 campaign and corroborated in 2012, MapSSS and CODI have generated very good results on the benchmark track. This suggests a strong correlation between the ranking in Benchmark and the ranking for MultiFarm test cases of type (ii), while there is, on the other hand, no (or only a very weak) correlation between results for test cases of type (i) and type (ii). For that reason, we only analyze in the following the results for test cases of type (i).

			Different ontologies (i)			Same ontologies (ii)		
	System	Runtime	Prec.	Fmeas.	Rec.	Prec.	Fmeas.	Rec.
Multilingual	AUTOMSv2	512.7	**.49**	**.36**	.10	**.69**	.24	.06
	GOMMA	35.0	**.29**	**.31**	**.36**	.63	**.38**	.29
	MEDLEY	76.5	.16	**.16**	.07	.34	.18	.09
	WeSeE	14.7	**.61**	**.41**	**.32**	**.90**	**.41**	.27
	Wmatch	1072.0	.22	.21	**.22**	.43	.17	.11
	YAM++	367.1	**.50**	**.40**	**.36**	**.91**	**.60**	**.49**
Non specific	AROMA	6.9				.31	.01	.01
	CODI	x	**.17**	.08	.02	**.82**	**.62**	**.50**
	Hertuda	23.5	.00	.01	1.00	.02	.03	**1.00**
	HotMatch	16.5	.00	.01	.00	.40	.04	.02
	LogMap	14.9	**.17**	.09	.02	.35	.03	.01
	LogMapLt	5.5	.12	.07	.02	.30	.03	.01
	MaasMatch	125.0	.02	.03	.14	.14	.14	.14
	MapSSS	17.3	.08	.09	.04	**.97**	**.66**	**.50**
	Optima	142.5	.00	.01	**.59**	.02	.03	**.41**

Table 7. MultiFarm aggregated results per matcher, for each type of matching task – types (i) and (ii). Runtime is measured in minutes (time for completing the 36×25 matching tasks). The top-5 values for each column are marked in bold-face.

6.4 Language specific results

Table 8 shows the results aggregated per language pair. For the sake of readability, we present only F-measure values. The reader can refer to the OAEI results web page for more detailed results on precision and recall.

As expected and already reported above, the systems that apply specific strategies to deal with multilingual matching labels outperform all other systems (overall F-measure for both cases): YAM++, followed by WeSeE, GOMMA, AUTOMSv2, Wmatch, and MEDLEY, respectively. Wmatch has the ability to deal with all pairs of languages, what is not the case for AUTOMSv2 and MEDLEY, specially for the pairs involving Chinese, Czech and Russian languages.

Most of the systems translating non-English ontology labels to English have better scores on pairs where English is present (by group of pairs, YAM++ is the typical case). This owes to the fact that multiple translations (pt $\rightarrow$ en and fr $\rightarrow$ en, for matching pt $\rightarrow$ fr, for instance) may result in more ambiguous translated concepts, which makes harder the process of finding correct correspondences. Furthermore, as somehow expected, good results are also obtained for pairs of languages having similarities on their vocabularies (es-pt and fr-pt, for instance). These two observations may explain the top F-measures of the specific multilingual methods: AUTOMSv2 (es-pt, en-es, de-nl, en-nl), GOMMA (en-pt, es-pt, cn-en, de-en), MEDLEY (en-fr, en-pt, cz-en, en-es), WeseE (en-es, es-fr, en-pt, es-pt, fr-pt), YAM++ (cz-en, cz-pt, en-pt). Wmatch has an interesting pair score, where Russian appears in the top F-measures: nl-ru, en-es, en-nl, fr-ru, es-ru. This may be explained by the use of Wikipedia multilingual inter-links, which are not limited to English or language similarities.

Pair	AUTOMSv2	CODI	GOMMA	Herduda	HotMatch	LogMap	LogMapLt	MaasMtch	MapSSS	MEDLEY	Optima	WeSeE	Wmatch	YAM++
cn cz			.34	.01				.01			.01	.29	.21	.35
cn de			.34	.01				.01			.01	.25	.08	.35
cn en			.41	.01				.00			.01	.31	.10	.41
cn es			.33	.01	.01			.00			.01	.34	.14	.20
cn fr			.35	.01	.01			.00			.01	.32	.09	.39
cn nl			.25	.01	.01			.00			.01	.23	.10	.34
cn pt			.31	.01				.01			.01	.30	.09	.35
cn ru			.27	.01	.01			.00			.01	.27	.13	.33
cz de		.10	.24	.01		.10	.09	.06	.07	.19	.01	.41	.24	.45
cz en		.07	.36	.01		.05	.04	.06	.08	.28	.01	.48	.24	.58
cz es		.11	.30	.01		.11	.11	.06	.11	.13	.01	.47	.25	.20
cz fr		.01	.16	.01	.01	.01	.01	.04	.01	.08	.01	.47	.20	.53
cz nl		.09	.21	.01		.04	.04	.07	.05	.09	.01	.48	.21	.55
cz pt		.15	.37	.01		.13	.13	.06	.12	.18	.01	.44	.12	.57
cz ru			.21	.01				.00			.01		.17	.49
de en	.38	.20	.41	.01		.22	.20	.06	.16	.27	.01	.39	.28	.52
de es	.35	.06	.35	.01		.12	.06	.06	.15	.15	.01	.41	.24	.20
de fr	.32	.04	.21	.01		.04	.04	.05	.13	.13	.01	.41	.25	.46
de nl	.39	.05	.24	.01		.04	.04	.06	.15	.12	.01	.37	.25	.40
de pt	.35	.08	.36	.01		.07	.07	.05	.06	.15	.01	.35	.22	.42
de ru			.33	.01	.01			.01			.01	.42	.26	.47
en es	.42	.04	.40	.01		.15	.04	.08	.18	.28	.01	.52	.30	.23
en fr	.31	.04	.36	.01	.01	.06	.04	.09	.13	.33	.01	.48	.27	.53
en nl	.39	.10	.38	.01		.08	.10	.09	.15	.24	.01	.49	.29	.53
en pt	.37	.08	.45	.01		.06	.06	.06	.07	.30	.01	.51	.26	.56
en ru			.34	.01				.01			.01	.43	.25	.47
es fr	.37	.01	.29	.01		.07	.01	.08	.06	.06	.01	.52	.24	.20
es nl	.38		.33	.01				.05	.01	.03	.01	.46	.27	.16
es pt	.44	.22	.44	.01		.24	.23	.11	.23	.22	.01	.51	.25	.25
es ru			.21	.01				.00			.01		.28	.19
fr nl	.27	.13	.21	.01	.01	.13	.12	.07	.11	.16	.01	.43	.28	.47
fr pt	.35		.32	.01	.01			.06	.02	.08	.01	.50	.23	.53
fr ru			.24	.01				.00			.01	.47	.28	.46
nl pt	.37	.04	.32	.01		.01	.01	.05	.02	.06	.01	.47	.20	.51
nl ru			.22	.01				.01			.01	.42	.31	.42
pt ru			.30	.01				.00			.01	.45	.17	.44

Table 8. MultiFarm results per pair of language, for the test cases of type (i). We distinguished empty alignments, represented by empty cells, from wrong ones.

For non-specific systems, though all of them cannot deal at all with Chinese and Russian languages, MapSSS, LogMap and CODI obtain better results. These system perform better for some specific pairs: MapSSS (es-pt, en-es, de-en), LogMap and LogMapL (es-pt, de-en, en-es, cz-pt), CODI (es-pt, de-en, cz-pt). From all these sys-

tems, the pairs es-pt and de-en are obtain better F-measures. Again, we can see that similarities in the language vocabulary have an important role in the matching task. On the other hand, although it is likely harder to find correspondences between cz-pt than es-pt, for some systems their best score include such combinations (cz-pt, for CODI and LogMapLt). This can be explained by the specific way systems combine their internal matching techniques (ontology structure, reasoning, coherence, linguistic similarities, etc).

6.5 Conclusions

We observe that specific methods for dealing with ontologies that are described in different languages, work much better than non specific systems. This is the expected behavior. However, the absolute results are still not very good, if compared to the top results of the original Conference dataset ($\cong$ .75 F-measure for the best matcher). For all specific multilingual methods, the techniques implemented in YAM++ generate the best alignments in terms of F measure ($\cong$ .53 overall F-measure for both types of matching tasks). YAM++ is followed by WeSeE and GOMMA, respectively. With the exception of Wmatch, all systems use English as pivot language.

Looking at the participation in the OAEI 2011.5 campaign, only 3 participants, out of 19 have used specific techniques. We counted this year with new systems implementing specific multilingual methods (seven out of 21). Although there is room for improvements to achieve the same level of compliance than in the original dataset, the increasing number of matchers dealing with multilingual matching is a sign that the field is progressing.

7 Library

This library track is a new track within the OAEI. Its challenge is to match two real-world thesauri: TheSoz (social sciences) and STW (economics). However, there has already been a library track from 2007 to 2009 [9; 3; 6] using different thesauri,[8] as well as other thesaurus tracks like the food track[9] and the environment track.[10] A common motivation is that these tracks use a real-world scenario, i.e., real thesauri. For us, it is still a motivation to develop a better understanding, how thesauri differ from ontologies and how these differences affect state-of-the-art ontology matchers. We hope that the community accepts the challenge and that subsequently significant improvements can be seen that push the quality of automatic alignments between thesauri. Furthermore, we will use the matching results as input for the maintainers of the reference alignment to improve the alignment. While a full manual evaluation of all matching results is certainly not feasible, this way we constantly improve the reference alignment and mitigate possible weaknesses and incompleteness.

[8] http://oaei.ontologymatching.org/2009/library/
[9] http://oaei.ontologymatching.org/2006/food/
[10] http://oaei.ontologymatching.org/2007/environment/

7.1 Test data

The library track uses two real-world thesauri, that are in many aspects comparable. They have roughly the same size, are both originally developed in German, are today both multilingual, both have English translations, and, most important, despite being from two different domains, they have huge overlapping areas. Not least, both are freely available in RDF using SKOS.[11]

STW The STW Thesaurus for Economics provides vocabulary on any economic subject: more than 6,000 standardized subject headings (skos:Concepts, with preferred labels in English and German) and 19,000 additional keywords (skos:altLabels) in both languages. The vocabulary was developed for indexing purposes in libraries and economic research institutions and includes technical terms used in law, sociology, or politics, and geographic names. The entries are richly interconnected by 16,000 skos:broader/narrower and 10,000 skos:related relations. An additional hierarchy of main categories provides a high level overview. The vocabulary is maintained on a regular basis by ZBW[12], the German National Library of Economics - Leibniz Centre for Economics, and has been translated into SKOS [24].

TheSoz The Thesaurus for the Social Sciences (TheSoz) serves as a crucial instrument for indexing documents and research information in the social sciences. It contains overall about 12,000 keywords, from which 8,000 are standardized subject headings (in English and German) and 4,000 additional keywords. The thesaurus covers all topics and sub-disciplines of the social sciences. Additionally terms from associated and related disciplines are included in order to support an accurate and adequate indexing process of interdisciplinary, practical-oriented and multi-cultural documents. The thesaurus is owned and maintained by GESIS[13], the Leibniz Institute for the Social Sciences, and is available in SKOS [31].

Reference Alignment An alignment between STW and TheSoz already exists and has been manually created by domain experts in the KoMoHe project [18]. However, it does not cover the changes and enhancements in both thesauri since 2006. It is available in SKOS with the different matching types SKOS:exactMatch, SKOS:broaderMatch and SKOS:narrowerMatch. Within the reference alignment, concepts of one thesaurus are aligned to more than one concept of the second thesaurus. Thus, we face a *n:m* mapping of the concepts. All in all, 4,285 TheSoz concepts and 2,320 STW concepts are aligned with 2,839 exact matches, 34 broader matches and 1,416 narrower matches. It is important to note that the reference alignment only contains alignments between the descriptors of both thesauri, i.e., the concepts that are actually used for document indexing. The upper part of the hierarchy consists of non-descriptor concepts (or categories) that are only used to organize the descriptors below them. We take this specialty into account as we only assess the generated alignments between descriptors and ignore alignments between non-descriptors. However, this might change in the future, as the

[11] http://www.w3.org/2004/02/skos/
[12] http://zbw.eu/index-e.html
[13] http://www.gesis.org/en/home

results of this track could be used to extend the reference alignment to the upper part of the hierarchy.

Transformation Most ontology matching systems taking part in the OAEI only work on OWL ontologies and are not (yet) ready to deal with the specialties of a thesaurus. To get first results and to lower the barrier of taking part in this challenge, we provide OWL versions of the thesauri, generated as follows:

`skos:concept`	→	`owl:class`
`skos:prefLabel`, `skos:altLabel`	→	`rdfs:label`
`skos:scopeNote`, `skos:notation`	→	`rdfs:comment`
`A skos:narrower B`	→	`B rdfs:subClassOf A`
`skos:broader`	→	`rdfs:subClassOf`
`skos:related`	→	`rdfs:seeAlso`

This transformation obviously is not lossless. First and foremost, within the ontology, it is not recognizable which label is the preferred one and which ones are alternative labels. Since matching systems mostly have to focus on the labels, this transformation might lead to suboptimal results. There are, however, more fundamental differences between ontologies and thesauri that we show in the next section.

SKOS vs. OWL Thesauri – and other, similar knowledge structures like classifications or taxonomies – are often called lightweight ontologies [29]. However, ontologies and thesauri fundamentally differ. This is also reflected by the fact that with SKOS a specific model for thesauri exists that is formulated in OWL. There, a `skos:Concept` is not an `owl:Class`. Concepts sometimes represent classes, for example the STW concept COMMODITIES. However, this is not true for every `skos:Concept`, e.g., the STW concept GERMANY is an instance, not a class.

Having a look at the subordinate concepts of COMMODITIES, they mostly indeed represent classes, like METALS – METAL PRODUCTS – RAZOR. Nevertheless, the relation in SKOS between these concepts is `skos:broader`, not `rdfs:subClassOf`. A subclass relationship states that if a class *B* is a subclass of a class *A*, then all instances of *B* will also be instances of *A*. Here, all metals are commodities, but not all metal products are metals: the razor consists partly of metal, but it is no metal.

Thesauri are created for a very specific purpose and are used in a predetermined way. This is inter alia reflected by the distinction of descriptors and non-descriptors. Only descriptors are assigned to publications during the indexation or classification. All non-descriptors serve as additional information to provide the correct context or to build up a proper hierarchy. Such a distinction typically does not exist in an ontology.

Very difficult for ontology matchers (not necessarily only automatic ones) is the quasi-synonymy of the describing labels for a concept. A `skos:altLabel` is often used to indicate subconcepts that should be subsumed under the concept in question to avoid extensive subclassing. As an example, the STW descriptor *14117-2* with the preferred English label "Tropical fruit" has German alternative labels like "pineapple",

"avocado", and "kiwi". In an (OWL) ontology, these alternative labels should be modeled as subclasses of the class TROPICAL FRUIT. In contrast, other alternative labels might really indicate alternative, synonymous terms for the preferred label.

At last, instead of arbitrary semantic relations that are part of an ontology, in thesauri, relations like `skos:related` or `compoundEquivalence` in TheSoz exist. They often contain information for the (manual) use of the thesaurus for indexing, i.e., which descriptor should be used in which case or how combinations of descriptors are to be used. Transferring them to ontological relations is not always possible and depends often on the single case.

It can be seen that the development of a thesaurus matcher is indeed a challenge that differs from ontology matching. Nevertheless, the commonalities between thesauri and ontologies are large enough to pave the way for further developments by means of current ontology matchers.

7.2 Experimental Setting

To compare the created alignments with the reference alignment, we use the Alignment API. For this first evaluation, we only included equivalence relations (`skos:exactMatch`).

All matching processes have been performed on a Debian machine with one 2.4GHz core and 7GB RAM allocated to each system. The evaluation has been executed by using SEALS technologies. For ServOMap, ServOMapLt and Optima, we used slightly adapted ontologies as input since they cannot handle URIs with the last part only consisting of numbers as it is the case in the official version. Each participating system uses the OWL version. We computed precision, recall and F-measure ($\beta = 1$) for each matcher. We only consider equivalence correspondences between two descriptors as non-descriptors are not included in the reference alignment. This filtering improves the precision ($\approx 8\%$) as well as the F-measure ($\approx 4\%$) for all systems. Moreover, we measured the runtime, the size of the created alignment and checked whether a 1:1 alignment has been created. To assess the results of the matchers, we developed three straightforward matching strategies, using the original SKOS version of the thesauri:

- *Matcher*$_{prefDE}$: Compares the German lower-case preferred labels and generates a correspondence if these labels are completely equivalent.
- *Matcher*$_{prefEN}$: Compares the English lower-case preferred labels and generates a correspondence if these labels are completely equivalent.
- *Matcher*$_{pref}$: Creates a correspondence, if either *Matcher*$_{prefDE}$ or *Matcher*$_{prefEN}$ or both create a correspondence.
- *Matcher*$_{allLabels}$: Creates a correspondences whenever at least one label (preferred or alternative, all languages) of an entity is equivalent to one label of another entity.

7.3 Results

All systems listed in Table 9 are sorted according to their F-measures. Altogether 13 of the 21 submitted matching systems were able to create an alignment. Three matching

System	Precision	F-measure	Recall	Time [s]	Size	1:1
Matcher$_{pref}$	0.820	0.720	0.642	75	2190	-
Matcher$_{prefDE}$	0.891	0.717	0.601	42	1885	-
Matcher$_{allLabels}$	0.544	0.677	0.896	735	4605	-
GOMMA	0.537	0.674	0.906	804	4712	-
ServOMapLt	0.654	0.670	0.687	45	2938	-
LogMap	0.688	0.665	0.644	95	2620	-
ServOMap	0.717	0.665	0.619	44	2413	$\surd$
YAM++	0.595	0.664	0.750	496	3522	-
LogMapLt	0.577	0.662	0.776	21	3756	-
Hertuda	0.465	0.619	0.925	14363	5559	-
WeSeE	0.612	0.609	0.607	144070	2774	$\surd$
HotMatch	0.645	0.608	0.575	14494	2494	$\surd$
Matcher$_{prefEN}$	0.808	0.569	0.439	36	1518	-
CODI	0.434	0.445	0.481	39869	3100	$\surd$
MapSSS	0.520	0.272	0.184	2171	989	$\surd$
AROMA	0.107	0.184	0.652	1096	17001	-
Optima	0.321	0.117	0.072	37457	624	-

Table 9. Results of the Library track.

systems (MaasMatch, MEDLEY, Wmatch) did not finish within the time frame of one week while five threw an exception (no heap space exception).

Of all these systems, GOMMA performs best in terms of F-measure, closely followed by ServOMapLt and LogMap. However, the precision and recall measures vary a lot across the three systems. Depending on the application, an alignment either achieving high precision or recall is preferable. If recall is in the focus, the alignment created by GOMMA is probably the best choice with a recall of about 90%. Other systems generate alignments with higher precision, e.g. ServOMap with over 70% precision, while mostly having significantly lower recall values (except for Hertuda).

From the results obtained by the matching strategies taking the different types of labels into account, we can see that a matching based on preferred labels only, outperforms other matching strategies. *Matcher*$_{pref}$ achieves the highest F-measure in these tests. The results of *Matcher*$_{prefDE}$ and *Matcher*$_{prefEN}$ provide an insight into the language characteristics of both thesauri and the reference alignments. *Matcher*$_{prefDE}$ achieves the highest precision value (nearly 90%), albeit with a recall of only 60%. Both thesauri as well as the reference alignment have been developed in Germany and focus on German terms. From the results of *Matcher*$_{prefEN}$, we can see the difference: precision and especially recall significantly decrease when only the preferred English labels are used. On the one hand, only about 80% of the found correspondences are correct and on the other hand, less than a half of all correspondences can be found this way. This can be a disadvantage for systems that use NLP techniques on English labels or rely on language-specific background knowledge like WordNet.

The high precision values of the $pref*$ matchers reflect the fact that the preferred labels are chosen specifically to unambiguously identify the concepts. Our interpretation is that the English translations are partly not as precise as the original German

terms (drop in precision) and not consistent regarding the English terminology (drop in recall).

In contrast, the *Matcher*$_{allLabels}$ achieves a quite high recall (90%) but a rather low precision (54%). This means that most but not all of the correspondences can be found by only having a look at equivalent labels. However, when following this idea, nearly a half of the found correspondences are incorrect. The rather high F-measure of *Matcher*$_{allLabels}$ is therefore misleading, as at least if the results would be used unchecked in an retrieval system, a higher precision would clearly be preferred over a higher recall. In this respect, matchers like ServOMap show better results. In any case, it can be seen that a matching system using the original SKOS version could achieve a better result. The information loss when converting SKOS to OWL really matters.

Concerning runtime, LogMap as well as ServOMap are quite fast with a runtime below 50 seconds. These values are comparable or even better (LogMapLt) than both strategies computing the equivalence between preferred labels. Thus, they are very effective in matching large ontologies while achieving very good results. Other matchers take several hours or even days and do not produce better alignments in terms of F-measure. By computing the correlation between F-measure and runtime, we notice a slightly negative correlation (-0.085) but the small amount of samples is not sufficient to make a significant statement. However, we can say for certain that a longer runtime does not necessarily lead to better results.

We further observe that the *n:m* reference alignment affects the results because some matching systems (ServOMap, WeSeE, HotMatch, CODI, MapSSS) only create 1:1 alignments and discard correspondences with entities that already occur in another correspondence. Whenever a system creates a lot of n:m correspondences, e.g., Hertuda and GOMMA, the recall significantly increases. This difference becomes clear when comparing ServOMapLt and ServOMap. Both systems are mostly based on the same methods but ServOMapLt does not use the 1:1 filtering. Consequently, its recall increases and its precision decreases.

Since the reference alignment has not been updated for about six years, it does not contain updates of both thesauri. Thus, new correct correspondences might be found by matching systems but they are indicated as incorrect because they are not included in the reference alignment. Therefore, we applied a manual evaluation to check whether matching systems found correct correspondences which are not included in the reference alignment at all. In turn, these information can help to improve the reference alignment.

The manual evaluation has been conducted by domain experts. Many newly detected correspondences, which have not been contained in the reference alignment yet, have been considered. By now, we only examined correspondences between descriptors as well as the ones that did not contain a term which is already matched in the reference alignment.

The matchers detected between 38 and 251 correct correspondences, which have not been in the reference alignment before. This includes especially terms, which hold a strong syntactical similarity or equivalence. But, some matching systems even detected difficult correspondences, e.g., between the German label for “automated production” (“Automatische Produktion”) and “CAM”, which has been identified by their associated

non-preferred labels. Furthermore, correspondences of geographical terms have been detected, but some of the matchers have not been able to distinguish between the terms for citizens of a country, their language or the country itself, although these differences can be derived from the structure of the thesauri.

But this manual evaluation exposed several issues, which can either be explained by the typical behavior of matching systems or by domain-specific differences inside the thesauri. There are similar terms inside TheSoz and STW, which are used in totally different contexts, e.g., the term "self-assessment". Even when considering the structure of both thesauri these differences are difficult to identify. In general, term similarities often led to wrong correspondences, which is not surprising at first. But, in turn syntactically equal terms have not been detected simultaneously in some cases. By now, we did not have the possibility to evaluate the matching systems with the improved reference alignment, but we plan to perform this additional evaluation soon.

7.4 Conclusion

Nevertheless, the newly detected correspondences determine already a useful result for the maintainers of the two thesauri. The correct correspondences can be added to the existing reference alignment, which is already applied in information portals for supporting search term recommendation and query expansion services among differently indexed databases. As all matching systems delivered exact matches for the correspondences, some of the wrong correspondences will be examined again in the future, whether other relationships like broader, narrower or related matches can be considered for those.

We expect further improvements, if the matchers are tailored more specifically to the library track, i.e., if they exploit the information found in the original SKOS version. A promising approach is also the use of additional knowledge, e.g., instance data – resources that are indexed with different thesauri [30].

This time, we collected the results of the matchers as a first survey and compared them to our simple string-matching strategy that takes advantage of the different types of labels. In future evaluations, we assume that better results can be achieved and that these strategies simply form a baseline.

8 Large biomedical ontologies

This track aims at finding alignments between the large and semantically rich biomedical ontologies FMA, SNOMED CT, and NCI, which contains 78,989, 306,591 and 66,724 classes, respectively.

8.1 Test data

UMLS Metathesaurus [2] has been selected as the basis for the track's reference alignments. UMLS is currently the most comprehensive effort for integrating independently-developed medical thesauri and ontologies, including FMA, SNOMED CT, and NCI. Although the standard UMLS distribution does not directly provide "alignments" (in the

OAEI sense) between the integrated ontologies, it is relatively straightforward to extract them from the information provided in the distribution files (see [17] for details).

It has been noticed, however, that although the creation of UMLS alignments combines expert assessment and auditing protocols they lead to a significant number of logical inconsistencies when integrated with the corresponding source ontologies [17].

To address this problem, we have considered two refinements of the UMLS alignments that do not lead to (many) unsatisfiable classes. These refinements have been generated using LogMap's repair facility [16] and the Alcomo debugging system [20].

The track has been split into three matching problems: FMA-NCI, FMA-SNOMED and SNOMED-NCI; and each matching problem in three tasks involving different fragments of the input ontologies.

FMA-NCI matching We have compared the results of the matching tools against both the original and refined UMLS alignment sets:

- Original UMLS alignments: 3,024 alignments ($\equiv$).
- Refined UMLS alignments:
  - LogMap's repair module : 2,898 alignments ($\equiv$, $\sqsubseteq$, $\sqsupseteq$).
  - Alcomo debbuging system: 2,819 alignments ($\equiv$).

Three tasks have been considered involving different fragments of FMA and NCI:

- *Task 1* consists of matching two (relatively small) modules of FMA (3,696 classes, 5%) and NCI (6,488 classes 10%).
- *Task 2* consists of matching two (relatively large) modules of FMA (28,861 classes, 37%) and NCI (25,591 classes, 38%).
- *Task 3* consists of matching the whole FMA and NCI ontologies.

FMA-SNOMED matching We have compared the results of the matching tools against both the original and refined UMLS alignment sets:

- Original UMLS alignments: 9,008 alignments ($\equiv$).
- Refined UMLS alignments:
  - LogMap's repair module : 8,111 alignments ($\equiv$, $\sqsubseteq$, $\sqsupseteq$).
  - Alcomo debbuging system: 8,132 alignments ($\equiv$).

Three tasks have been considered involving different fragments of FMA and SNOMED:

- *Task 4* consists of matching two (relatively small) modules of FMA (10,157 classes, 13%) and SNOMED (13,412 classes, 5%).
- *Task 5* consists of matching two (relatively large) modules of FMA (50,523 classes, 64%) and SNOMED (122,464 classes, 40%).
- *Task 6* consists of matching the whole FMA and SNOMED ontologies.

SNOMED-NCI matching We have compared the results of the matching tools against both the original and refined UMLS alignment sets:

- Original UMLS alignments: 18,844 alignments ($\equiv$).
- Refined UMLS alignments:
  - LogMap's repair module : 18,324 alignments ($\equiv$, $\sqsubseteq$, $\sqsupseteq$).

Note that, at the time of creating the datasets, we could not compute a refined UMLS alignment set with Alcomo. The new version of Alcomo, however, has shown to be able to cope with SNOMED-NCI. Three tasks have been considered involving different fragments of SNOMED and NCI:

- *Task 7* consists of matching two (relatively small) modules of SNOMED (51,128 classes, 17%) and NCI (23,958 classes, 36%).
- *Task 8* consists of matching two (relatively large) modules of SNOMED (122,464 classes, 40%) and NCI (49,795 classes, 75%).
- *Task 9* consists of matching the whole SNOMED and NCI ontologies.

8.2 Results

We have run the evaluation in a high performance server with 16 CPUs and allocating 15GB RAM. In total, 15 out of 23 participating systems/configurations have been able to cope with at least one of the tasks of the track matching problem. Optima and MEDLEY failed to complete the smallest task with a time out of 24 hours, while OMR, OntoK, ASE and WeSeE, threw an Exception during the matching process. CODI was evaluated in a different setting using only 7GB and threw an exception related to insufficient memory when processing the smallest matching task. TOAST was not evaluated since it was only configured for the Anatomy track and it required a complex installation. LogMapLt, a very fast string matcher, has been used as baseline.

Note that GOMMA has also been evaluated with a configuration that exploits specialized background knowledge based on the UMLS Metathesaurus (GOMMA_{bk}). GOMMA_{bk} exploits the alignments $\mathcal{O}_1$-UMLS and UMLS-$\mathcal{O}_2$ and applies alignment composition techniques. LogMap, MaasMatch and YAM++ also use different kinds of background knowledge. LogMap uses normalisations and spelling variants from the domain specific resource UMLS Lexicon[14]. YAM++ and MaasMatch use the general purpose background knowledge provided by WordNet[15].

LogMap has also been evaluated with two configurations. LogMap's default algorithm computes an estimation of the overlapping between the input ontologies before the matching process, while the variant $\text{LogMap}_{\text{noe}}$ has this feature deactivated.

Precision and recall in Tables 11-13 average the obtained results with respect to the reference alignments. Systems have been ordered in terms of the average F-measure.

[14] `http://www.nlm.nih.gov/pubs/factsheets/umlslex.html`
[15] `http://wordnet.princeton.edu/`

Alignment coherence We have evaluated the coherence of the generated alignments and we have reported (1) number of unsatisfiabilities when reasoning[16] with the input ontologies together with the computed alignments, (2) the degree of unsatisfiable classes with respect to the size of the merged ontology (based on the Unsatisfiability Measure proposed in [22]), and (3) an approximation of the root unsatisfiability. The root unsatisfiability aims at providing a more precise amount of errors, since many of the unsatisfiabilities may be derived (i.e., a subclass of an unsatisfiable class will also be reported as unsatisfiable). The provided approximation is based on LogMap's (incomplete) repair facility and shows the number of classes that this facility needed to repair in order to solve (most of) the unsatisfiabilities [16].

LogMap and its variant $\text{LogMap}_{\text{noe}}$ were the unique systems generating an almost clean output. Tables 11-13 shown that even the most precise alignment sets may lead to a huge amount of unsatisfiable classes. This proves the importance of using techniques to assess the coherence of the generated alignments.

Note that, LogMap may fail to detect and repair unsatisfiable classes which are outside the computed overlapping (i.e. ontology fragments) between the input ontologies. Thus, $\text{LogMap}_{\text{noe}}$ provides, in general, a cleaner output than LogMap.

Runtimes Table 10 shows which systems were able to complete each of the matching tasks in less than 24 hours and the required computation times. Systems have been ordered with respect to the number of completed tasks and total time required to complete them. The last column reports the number of tasks that a system could complete. For example, only eight systems were able to complete all nine tasks. The last row shows the number of systems that could finish each of the tasks. The tasks involving larger ontology sizes were completed by only 8-10 systems. Furthermore, the tasks involving SNOMED were also harder with respect to both computation times and the number of systems that completed the tasks.

The runtimes were, in general, positive. For example, 7 systems completed Task 3 in less than 5 minutes. Additionally, the computation times for these systems increased "smoothly" with respect to the size of the input ontologies.

FMA-NCI matching Table 11 summarizes the results for the three tasks in the FMA-NCI matching problem. GOMMA_{bk} provided the best results in terms of F-measure in Task 1, whereas YAM++ in Tasks 2 and 3. GOMMA_{bk} also obtained the best results in terms of recall in all three tasks, while ServOMap computed the most precise alignments. Overall, the results were very positive and 7 systems obtained an F-measure greater than 0.80 in all three tasks.

LogMap and $\text{LogMap}_{\text{noe}}$ provided the same results in Task 1 since the input ontologies are already small fragments of FMA and NCI and thus, the overlapping estimation performed by LogMap did not have any impact.

[16] We have used HermiT [23] in the FMA-NCI and FMA-SNOMED matching problems. In the SNOMED-NCI matching problem we have estimated the number of unsatisfiable classes with the Dowling-Gallier algorithm for propositional Horn satisfiability [5] (implemented in LogMap's repair facility) since no OWL 2 reasoner is known to cope with the integration of SNOMED and NCI via alignments [15].

System	FMA-NCI			FMA-SNOMED			SNOMED-NCI			#
	Task 1	Task 2	Task 3	Task 4	Task 5	Task 6	Task 7	Task 8	Task 9	
LogMapLt	8	29	55	14	96	171	54	104	178	9
ServOMapL	20	95	251	39	234	517	147	363	738	9
ServOMap	25	98	204	46	315	532	153	282	654	9
LogMap	18	77	131	65	484	612	221	514	955	9
$\text{LogMap}_{\text{noe}}$	18	74	206	63	521	791	211	575	1,505	9
GOMMA	26	69	217	54	437	1,994	197	527	1,820	9
GOMMA_{bk}	26	83	231	148	636	1,893	226	638	1,940	9
YAM++	78	245	1,304	326	3,780	23,900	1,901	6,127	30,155	9
AROMA	63	7,538	-	51,191	62,801	-	15,624	-	-	5
MapSSS	561	30,575	-	3,129	-	-	27,381	-	-	4
Hertuda	3,327	-	-	17,625	-	-	-	-	-	2
HotMatch	4,271	-	-	31,718	-	-	-	-	-	2
MaasMatch	27,157	-	-	-	-	-	-	-	-	1
AUTOMSv2	62,407	-	-	-	-	-	-	-	-	1
Wmatch	65,399	-	-	-	-	-	-	-	-	1
Completed	**15**	**10**	**8**	**12**	**9**	**8**	**10**	**8**	**8**	**88**

Table 10. System runtimes (s) and task completion.

In Task 1, our baseline also provided very good results in terms of F-measure and outperformed 8 of the participating systems. MaasMatch and Hertuda provided competitive results in terms of recall, but the low precision damaged the final F-measure. MapSSS and AUTOMSv2 provided a set of alignments with high precision, however, the F-measure was damaged due to the low recall of their alignments.

Efficiency in Task 2 and Task 3 have decreased considerably with respect to Task 1. This is mostly due to the fact that larger ontologies also involves more possible candidate alignments and it is harder to keep high precision values without damaging recall, and vice versa.

FMA-SNOMED matching Table 12 summarizes the results for the three tasks in the FMA-SNOMED matching problem. GOMMA_{bk} provided the best results in terms of F-measure in Task 4, whereas ServOMap in Tasks 5 and 6. GOMMA_{bk} also obtained the best results in terms of recall in all three tasks, while LogMapLt computed the most precise alignments in Task 4 and ServOMapL in Tasks 5 and 6.

Overall, the results were less positive than in the FMA-NCI matching tasks and only 5 systems obtained an F-measure greater than 0.70 in all three tasks. Furthermore, 6 systems (including our baseline) failed to provide a recall higher than 0.4. Thus, matching FMA against SNOMED represents a significant leap in complexity with respect to the FMA-NCI matching problem.

As in the FMA-NCI matching problem, efficiency also decreases as the ontology size increases. The most important variations were suffered by GOMMA_{bk} and GOMMA, where their average precision decreased from 0.893 and 0.875 (Task 4) to 0.571 and 0.389 (Task 5), respectively. This is an interesting fact, since the background knowledge used by GOMMA_{bk} could not avoid the decrease in precision while keeping the highest recall.

SNOMED-NCI matching Table 13 summarizes the results for the three matching tasks in the SNOMED-NCI matching problem. $\text{LogMap}_{\text{noe}}$ provided the best results in terms of both recall and F-measure in Tasks 7 and 8 while YAM++ obtained the

System	Time (s)	Size	Average			Incoherence		
			P	F	R	All Unsat.	Degree	Root Unsat.
Task 1: small FMA and NCI fragments								
$GOMMA_{bk}$	26	2,843	0.94	0.92	0.91	6,204	61%	193
YAM++	78	2,614	0.96	0.91	0.86	2,352	23%	92
$LogMap/LogMap_{noe}$	18	2,740	0.93	0.90	0.88	2	0,02%	0
GOMMA	26	2,626	0.95	0.90	0.86	2,130	21%	127
ServOMapL	20	2,468	0.96	0.88	0.82	5,778	57%	79
LogMapLt	8	2,483	0.95	0.87	0.81	2,104	21%	116
ServOMap	25	2,300	0.97	0.86	0.77	5,597	55%	50
HotMatch	4,271	2,280	0.96	0.84	0.75	285	3%	65
Wmatch	65,399	3,178	0.79	0.82	0.86	3,168	31%	482
AROMA	63	2,571	0.86	0.80	0.76	7,196	70%	421
Hertuda	3,327	4,309	0.58	0.69	0.86	2,675	26%	277
MaasMatch	27,157	3,696	0.61	0.68	0.78	9,598	94%	3,113
AUTOMSv2	62,407	1,809	0.80	0.62	0.50	5,346	52%	392
MapSSS	561	1,483	0.84	0.57	0.43	565	6%	94
Task 2: big FMA and NCI fragments								
YAM++	245	2,688	0.90	0.87	0.83	22,402	35%	102
ServOMapL	95	2,640	0.89	0.85	0.81	22,315	35%	143
GOMMA	69	2,810	0.86	0.84	0.83	2,398	4%	116
$GOMMA_{bk}$	83	3,116	0.81	0.84	0.87	4,609	8%	146
$LogMap_{noe}$	74	2,663	0.87	0.83	0.80	5	0.01%	0
LogMap	77	2,656	0.87	0.83	0.79	5	0.01%	0
ServOMap	98	2,413	0.91	0.83	0.76	21,688	34%	86
LogMapLt	29	3,219	0.73	0.77	0.81	12,682	23%	443
AROMA	7,538	3,856	0.54	0.61	0.69	20,054	24%	1600
MapSSS	30,575	2,584	0.38	0.36	0.34	21,893	40%	358
Task 3: whole FMA and NCI ontologies								
YAM++	1,304	2,738	0.89	0.86	0.83	50,550	29%	141
GOMMA	217	2,843	0.85	0.84	0.83	5,574	4%	139
ServOMapL	251	2,700	0.87	0.84	0.81	50,334	28%	164
$GOMMA_{bk}$	231	3,165	0.80	0.83	0.87	12,939	9%	245
$LogMap_{noe}$	206	2,646	0.87	0.83	0.79	9	0.01%	0
LogMap	131	2,652	0.86	0.82	0.78	9	0.01%	0
ServOMap	204	2,465	0.89	0.82	0.76	48,743	27%	114
LogMapLt	55	3,466	0.68	0.74	0.81	26,429	9%	778

Table 11. Results for the FMA-NCI matching problem.

best results in Task 9. $GOMMA_{bk}$ obtained the best recall in Task 9 and ServOMap generated the most precise alignments in all three tasks.

As in the previous matching problems, efficiency decreases as the ontology size increases. For example, in Task 9 none of the systems could reach an F-measure of 0.7, while seven systems (including our baseline) exceeded this value in Task 7.

8.3 Conclusions

Although the proposed matching tasks represented a significant leap in complexity with respect to the tasks in previous campaigns, the obtained results have been very promising and eight systems completed all matching tasks with very competitive results.

There is, however, plenty of room for improvement: (1) most of the participating systems disregard the coherence of the generated alignments; (2) the size of the input ontologies should not significantly affect efficiency; and (3) recall in the tasks involving SNOMED should be improved while keeping the current precision values.

The alignment coherence measure was the weakest point of the systems participating in this track. As shown in Tables 11-13, even highly precise alignment sets may

System	Time (s)	Size	Average			Incoherence		
			P	F	R	All Unsat.	Degree	Root Unsat.
Task 4: small FMA and SNOMED fragments								
$GOMMA_{bk}$	148	8,598	0.89	0.90	0.91	13,685	58%	4,674
ServOMapL	39	6,346	0.92	0.79	0.69	10,584	45%	3,056
YAM++	326	6,421	0.91	0.79	0.69	14,534	62%	3,150
$LogMap_{noe}$	63	6,363	0.91	0.78	0.69	0	0%	0
LogMap	65	6,164	0.91	0.77	0.67	2	0.01%	2
ServOMap	46	6,008	0.92	0.76	0.66	8,165	35%	2,721
GOMMA	54	3,667	0.88	0.53	0.38	2,058	9%	206
MapSSS	3,129	3,458	0.75	0.44	0.31	9,084	39%	389
AROMA	51,191	5,227	0.53	0.40	0.33	21,083	89%	2,296
HotMatch	31,718	2,139	0.84	0.34	0.21	907	4%	104
LogMapLt	14	1,645	0.94	0.31	0.18	773	3%	21
Hertuda	17,625	3,051	0.56	0.30	0.20	1,020	4%	47
Task 5: big FMA and SNOMED fragments								
ServOMapL	234	6,563	0.88	0.77	0.69	55,970	32%	1,192
ServOMap	315	6,272	0.88	0.75	0.65	143,316	83%	1,320
YAM++	3,780	7,003	0.82	0.75	0.68	69,345	40%	1,360
$LogMap_{noe}$	521	6,450	0.84	0.73	0.64	0	0%	0
LogMap	484	6,292	0.83	0.71	0.62	0	0%	0
$GOMMA_{bk}$	636	12,614	0.57	0.68	0.86	75,910	44%	3,344
GOMMA	437	5,591	0.39	0.31	0.26	7,343	4%	480
AROMA	62,801	2,497	0.66	0.30	0.20	54,459	31%	271
LogMapLt	96	1,819	0.85	0.30	0.18	2,994	2%	24
Task 6: whole FMA and SNOMED ontologies								
ServOMapL	517	6,605	0.88	0.77	0.69	99,726	26%	2,862
ServOMap	532	6,320	0.87	0.75	0.65	273,242	71%	2,617
YAM++	23,900	7,044	0.81	0.74	0.68	106,107	28%	3,393
LogMap	612	6,312	0.83	0.71	0.62	10	0.003%	0
$LogMap_{noe}$	791	6,406	0.82	0.71	0.62	10	0.003%	0
$GOMMA_{bk}$	1,893	12,829	0.56	0.68	0.86	119,657	31%	5,289
LogMapLt	171	1,823	0.85	0.30	0.18	4,938	1%	37
GOMMA	1,994	5,823	0.35	0.29	0.24	10,752	3%	609

Table 12. Results for the FMA-SNOMED matching problem.

lead to a huge number of unsatisfiable classes. Thus, the use of techniques to assess mapping coherence is critical. Unfortunately, only a few systems in OAEI 2012 have successfully used such techniques. In future campaigns, this aspect should not be neglected. Developers can reuse available state-of-the-art mapping debugging techniques such as the implemented in Alcomo [20] or in LogMap [16].

The UMLS-based reference alignments may contain errors and may also be incomplete. Thus, in order to turn the current reference alignments into a agreed-upon gold standard, expert assessment is required, which is almost unfeasible for large alignment sets. In this track, we have opted to move towards a "silver standard" by "harmonising" the outputs of different matching tools over the different matching tasks (see `http://www.cs.ox.ac.uk/isg/projects/SEALS/oaei/harmonisation/oaei2012_harmo.html` for details).

System	Time (s)	Size	Average			Incoherence		
			P	F	R	All Unsat.	Degree	Root Unsat.
Task 7: small SNOMED and NCI fragments								
$LogMap_{noe}$	211	13,525	0.90	0.75	0.65	0	0%	0
LogMap	221	13,454	0.90	0.75	0.65	0	0%	0
$GOMMA_{bk}$	226	12,294	0.94	0.75	0.62	48,681	65%	863
YAM++	1,901	11,961	0.95	0.74	0.61	50,089	67%	471
ServOMapL	147	11,730	0.95	0.74	0.60	62,367	83%	657
ServOMap	153	10,829	0.97	0.71	0.56	51,020	68%	467
LogMapLt	54	10,947	0.95	0.70	0.56	61,269	82%	801
GOMMA	197	10,555	0.94	0.68	0.53	42,813	57%	851
AROMA	15,624	11,783	0.85	0.66	0.54	70,491	94%	1,286
MapSSS	27,381	9,608	0.79	0.54	0.41	46,083	61%	794
Task 8: big SNOMED and NCI fragments								
$LogMap_{noe}$	575	13,184	0.88	0.73	0.62	0	0%	0
YAM++	6,127	13,083	0.86	0.71	0.61	104,492	61%	618
ServOMapL	363	12,784	0.86	0.70	0.59	136,909	79%	1,101
LogMap	514	12,142	0.87	0.69	0.57	3	0.002%	2
ServOMap	282	11,632	0.89	0.69	0.56	110,253	64%	820
$GOMMA_{bk}$	638	15,644	0.72	0.66	0.61	116,451	68%	2,741
LogMapLt	104	12,741	0.81	0.66	0.56	131,073	76%	2,201
GOMMA	527	12,320	0.80	0.63	0.53	96,945	56%	1,621
Task 9: whole SNOMED and NCI ontologies								
YAM++	30,155	14,103	0.79	0.68	0.60	238,593	64%	979
ServOMapL	738	13,964	0.79	0.68	0.59	286,790	77%	1,557
LogMap	955	13,011	0.81	0.67	0.57	16	0.004%	10
$LogMap_{noe}$	1,505	13,058	0.81	0.67	0.57	0	0%	0
ServOMap	654	12,462	0.83	0.67	0.56	230,055	62%	1,546
$GOMMA_{bk}$	1,940	17,045	0.66	0.64	0.61	239,708	64%	4,297
LogMapLt	178	14,043	0.74	0.63	0.56	305,648	82%	3,160
GOMMA	1,820	13,693	0.71	0.61	0.53	215,959	58%	2,614

Table 13. Results for the SNOMED-NCI matching problem.

9 Instance matching

The instance matching track aims at evaluating the performance of different matching tools on the task of matching RDF individuals which originate from different sources but describe the same real-world entity. Data interlinking is known under many names according to various research communities: equivalence mining, record linkage, object consolidation and coreference resolution to mention the most used ones. In each case, these terms are used for the task of finding equivalent entities in or across data sets [14]. As the quantity of data sets published on the Web of data dramatically increases, the need for tools helping to interlink resources becomes more critical. It is particularly important to maximize the automation of the interlinking process in order to be able to follow this expansion.

Unlike the other tracks, the instance matching track specifically focuses on an ontology ABox. However, the problems which have to be resolved in order to correctly match instances can originate at the schema level (use of different properties and classification schemas) as well as at the data level, e.g., different formats of values. This year, the track included two tasks. The first task, called *Sandbox*, is a simple dataset that has been specifically conceived to provide a examples of some specific matching problems highlighted (like name spelling and other controlled variations). This is intended to serve as a test for those tools that are in an initial phase of their development

process and/or for tools that are facing very focused tasks, like for example person name matching. The second one, called *IIMB* is an OWL-based dataset that is automatically generated by introducing a set of controlled transformations in an initial OWL Abox.

The list of participants to the instance matching track is shown in Table 14.

Dataset	**LogMap**	**LogMap_lite**	**SBUEI**	**semsim**
Sandbox	√	√	√	
IIMB	√	√	√	√

Table 14. Participants in the instance matching track.

9.1 Sandbox

The dataset used for the Sandbox task has been automatically generated by extracting data from Freebase, an open knowledge base that contains information about 11 million real objects including movies, books, TV shows, celebrities, locations, companies and more. Data has been extracted in JSON through the Freebase JAVA API[17]. Sandox is a collection of OWL files consisting of 31 concepts, 36 object properties, 13 data properties and 375 individuals divided into 10 test cases[18]. In order to provide simple matching challenges mainly conceived for systems in their initial developing phase, we limited the way data are transformed from the original Abox to the test cases. In particular, we introduced only changes in data format (misspelling, errors in text, etc.).

Sandbox results An overview of the precision, recall and F_1-measure results of the Sandbox task is shown in Table 15.

test	Precision	Recall	F_1-measure
LogMap	0.94	0.94	0.94
LogMap_lite	0.95	0.89	0.92
SBUEI	0.95	0.98	0.96

Table 15. Results of the Sandbox task.

As expected, all the participating systems obtained very good results for the simple tests provided by the Sandbox task. This result confirms that the currently available systems for instance matching provide efficient facilities for data matching when dealing with simple errors and syntactic heterogeneities.

9.2 IIMB

The IIMB task is focused on two main goals:

1. to provide an evaluation data set for various kinds of data transformations, including value transformations, structural transformations and logical transformations;

[17] `http://code.google.com/p/freebase-java/`

[18] DL expressivity of ontologies is $\mathcal{ALHI}(D)$

2. to cover a wide spectrum of possible techniques and tools.

ISLab Instance Matching Benchmark (IIMB), that has been generated using the SWING tool [13]. Participants were requested to find the correct correspondences among individuals of the first knowledge base and individuals of the other one. An important task here is that some of the transformations require automatic reasoning for finding the expected alignments.

IIMB is composed of a set of test cases, each one represented by a set of instances, i.e., an OWL ABox, built from an initial data set of real linked data extracted from the web. Then, the ABox is automatically modified in several ways by generating a set of new ABoxes, called *test cases*. Each test case is produced by transforming the individual descriptions in the reference ABox in new individual descriptions that are inserted in the test case at hand. The goal of transforming the original individuals is twofold: on one side, we provide a simulated situation where data referring to the same objects are provided in different data sources; on the other side, we generate different data sets with a variable level of data quality and complexity. IIMB provides transformation techniques supporting modifications of data property values, modifications of number and type of properties used for the individual description, and modifications of the individuals classification. The first kind of transformations is called *data value transformation* and it aims at simulating the fact that data expressing the same real object in different data sources may be different because of data errors or because of the usage of different conventional patterns for data representation. The second kind of transformations is called *data structure transformation* and it aims at simulating the fact that the same real object may be described using different properties/attributes in different data sources. Finally, the third kind of transformations, called *data semantic transformation*, simulates the fact that the same real object may be classified in different ways in different data sources.

The 2012 edition has been created by exploiting the same OWL source used for the Sandbox task. The main difference is that we introduced in IIMB a large set of data transformations. In particular, test cases from 0 to 20 contain changes in data format (misspelling, errors in text, etc); test cases 21 to 40 contain changes in structure (properties missing, RDF triples changed); 41 to 60 contain logical changes (class membership changed, logical errors); finally, test cases 61 to 80 contain a mix of the previous.

IIMB results An overview of the precision, recall and F_1-measure results per set of tests of the IIMB subtrack is shown in Table 16. A precision-recall graph visualization is shown in Figure 5.

As a general comment, we can conclude that all the four systems participating in this edition of the instance matching track obtained good results, both in terms of precision and recall. Table 16 suggests that the most challenging tasks in IIMB this year were those included in test cases 061-080, which provides a combination of different data transformations, ranging from syntactic to semantic transformations. According to this conclusion, we will better investigate the problem of dealing with the problem of combining data transformations in order to generate a new challenging dataset for instance matching evaluation.

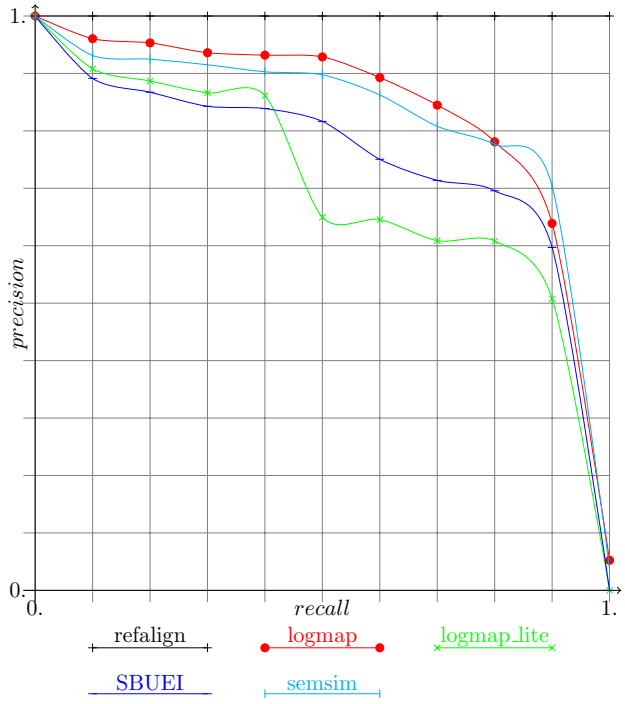

Fig. 5. Precision/recall of the systems participating in the IIMB subtrack.

test	001–020			021–040			041–060			061–080		
	P	F	R	P	F	R	P	F	R	P	F	R
LogMap	.94	.90	.87	.93	.96	1.0	.93	.96	1.0	.95	.86	.79
LogMap_lite	.95	.78	.66	.93	.95	.97	.74	.84	.98	.77	.73	.69
SBUEI	.96	.97	.98	.97	.98	.99	.91	.90	.89	.58	.53	.48
semsim	.93	.93	.93	.91	.91	.91	.94	.94	.94	.66	.66	.66

Table 16. Results of the IIMB subtrack.

10 Lesson learned and suggestions

There are, this year, very few comments about the evaluation execution:

A) This year indicated again that requiring participants to implement a minimal interface was not a strong obstacle to participation. Moreover, the community seems to get used to the SEALS technology introduced for OAEI 2011. This might be one of the reasons for an increasing participation.
B) We have not delivered any comparative results prior to the deadline for submitting the papers that contain the systems description. This procedure was motivated by avoiding a focus on competitive aspects of OAEI. However, several tool developers have complained about this procedure, because they wanted to include such results in their papers.
C) Last years we reported that we had many new participants. The same trend can be observed for 2012.
D) Again, given the high number of publications on data interlinking, it is surprising to have so few participants to the instance matching tracks.

11 Conclusions

These year, both in OAEI 2011.5 and 2012, some tracks have focused on scalability and runtime measurement. The low number of systems that could generate results for the Anatomy track was an uncommon result in 2011. However, in 2012 many more systems could generate results for the Anatomy track. Moreover, many systems could also generate results for the Library and the Large Biomed track that is concerned with significantly larger test cases.

Compared to the previous years, we observed a significant improvements of runtimes. Matching systems are becoming more robust and also more efficient with respect to runtimes. In particular, these improvements are more general compared to increased precision and recall scores. We dare thinking that this improvement has been steered by OAEI efforts towards more challenging test sets.

There is a high variance in runtimes and there is no correlation between runtime and quality of the generated results. This is a result that we already observed in 2011.

There has been a considerable increase in the number of participants implementing specific techniques for dealing with the task of matching ontologies in different natural languages (seven participants in 2012, three in OAEI 2011.5). Although there is room for improvements to achieve the same level of compliance than in the original OntoFarm dataset, this increase is a sign that the field is progressing.

All participants have provided a description of their systems and their experience in the evaluation. These OAEI papers, like the present one, have not been peer reviewed. However, they are full contributions to this evaluation exercise and reflect the hard work and clever insight people put in the development of participating systems. Reading the papers of the participants should help people involved in ontology matching to find what makes these algorithms work and what could be improved. Sometimes participants offer alternate evaluation results.

The Ontology Alignment Evaluation Initiative will continue these tests by improving both test cases and testing methodology for being more accurate. Further information can be found at:

`http://oaei.ontologymatching.org`.

Acknowledgements

We warmly thank the participants of this campaign. We know that they have worked hard for having their matching tools executable in time and they provided insightful papers presenting their experience. The best way to learn about the results remains to read the following papers.

We would like to thank Andreas Oskar Kempf from GESIS for the manual evaluation of the new detected correspondences. We thank Jan Noessner for providing data in the process of constructing the IIMB data set. We are grateful to Martin Ringwald and Terry Hayamizu for providing the reference alignment for the anatomy ontologies and thank Elena Beisswanger for her thorough support on improving the quality of the data set.

We also thank the other members of the Ontology Alignment Evaluation Initiative steering committee: Yannis Kalfoglou (Ricoh laboratories, UK), Miklos Nagy (The Open University (UK), Natasha Noy (Stanford University, USA), Yuzhong Qu (Southeast University, CN), York Sure (Leibniz Gemeinschaft, DE), Jie Tang (Tsinghua University, CN), George Vouros (University of the Aegean, GR).

José Luis Aguirre, Bernardo Cuenca Grau, Jérôme Euzenat, Ernesto Jimenez-Ruiz, Christian Meilicke, Heiner Stuckenschmidt and Cássia Trojahn dos Santos have been partially supported by the SEALS (IST-2009-238975) European project.

Ondřej Šváb-Zamazal has been supported by the CSF grant P202/10/0761.

Ernesto and Bernardo have been partially supported by the Royal Society, EPSRC project LogMap.

References

1. Benhamin Ashpole, Marc Ehrig, Jérôme Euzenat, and Heiner Stuckenschmidt, editors. *Proc. of the K-Cap Workshop on Integrating Ontologies*, Banff (Canada), 2005.
2. Olivier Bodenreider. The unified medical language system (UMLS): integrating biomedical terminology. *Nucleic Acids Research*, 32:267–270, 2004.
3. Caterina Caracciolo, Jérôme Euzenat, Laura Hollink, Ryutaro Ichise, Antoine Isaac, Véronique Malaisé, Christian Meilicke, Juan Pane, Pavel Shvaiko, Heiner Stuckenschmidt, Ondrej Sváb-Zamazal, and Vojtech Svátek. Results of the ontology alignment evaluation initiative 2008. In *Proc. 3rd International Workshop on Ontology Matching (OM) collocated with ISWC*, pages 73–120, Karlsruhe (Germany), 2008.
4. Jérôme David, Jérôme Euzenat, François Scharffe, and Cássia Trojahn dos Santos. The alignment API 4.0. *Semantic web journal*, 2(1):3–10, 2011.
5. William F. Dowling and Jean H. Gallier. Linear-time algorithms for testing the satisfiability of propositional Horn formulae. *J. Log. Prog.*, 1(3):267–284, 1984.
6. Jérôme Euzenat, Alfio Ferrara, Laura Hollink, Antoine Isaac, Cliff Joslyn, Véronique Malaisé, Christian Meilicke, Andriy Nikolov, Juan Pane, Marta Sabou, François Scharffe,

Pavel Shvaiko, Vassilis Spiliopoulos, Heiner Stuckenschmidt, Ondrej Sváb-Zamazal, Vojtech Svátek, Cássia Trojahn dos Santos, George Vouros, and Shenghui Wang. Results of the ontology alignment evaluation initiative 2009. In *Proc. 4th Workshop on Ontology Matching (OM) collocated with ISWC*, pages 73–126, Chantilly (USA), 2009.

7. Jérôme Euzenat, Alfio Ferrara, Christian Meilicke, Andriy Nikolov, Juan Pane, François Scharffe, Pavel Shvaiko, Heiner Stuckenschmidt, Ondrej Sváb-Zamazal, Vojtech Svátek, and Cássia Trojahn dos Santos. Results of the ontology alignment evaluation initiative 2010. In Pavel Shvaiko, Jérôme Euzenat, Fausto Giunchiglia, Heiner Stuckenschmidt, Ming Mao, and Isabel Cruz, editors, *Proc. 5th ISWC workshop on ontology matching (OM) collocated with ISWC, Shanghai (China)*, pages 85–117, 2010.
8. Jérôme Euzenat, Alfio Ferrara, Robert Willem van Hague, Laura Hollink, Christian Meilicke, Andriy Nikolov, François Scharffe, Pavel Shvaiko, Heiner Stuckenschmidt, Ondrej Sváb-Zamazal, and Cássia Trojahn dos Santos. Results of the ontology alignment evaluation initiative 2011. In Pavel Shvaiko, Isabel Cruz, Jérôme Euzenat, Tom Heath, Ming Mao, and Christoph Quix, editors, *Proc. 6th ISWC workshop on ontology matching (OM), Bonn (DE)*, pages 85–110, 2011.
9. Jérôme Euzenat, Antoine Isaac, Christian Meilicke, Pavel Shvaiko, Heiner Stuckenschmidt, Ondrej Svab, Vojtech Svatek, Willem Robert van Hage, and Mikalai Yatskevich. Results of the ontology alignment evaluation initiative 2007. In *Proc. 2nd International Workshop on Ontology Matching (OM) collocated with ISWC*, pages 96–132, Busan (Korea), 2007.
10. Jérôme Euzenat, Christian Meilicke, Pavel Shvaiko, Heiner Stuckenschmidt, and Cássia Trojahn dos Santos. Ontology alignment evaluation initiative: six years of experience. *Journal on Data Semantics*, XV:158–192, 2011.
11. Jérôme Euzenat, Malgorzata Mochol, Pavel Shvaiko, Heiner Stuckenschmidt, Ondrej Svab, Vojtech Svatek, Willem Robert van Hage, and Mikalai Yatskevich. Results of the ontology alignment evaluation initiative 2006. In *Proc. 1st International Workshop on Ontology Matching (OM) collocated with ISWC*, pages 73–95, Athens, Georgia (USA), 2006.
12. Jérôme Euzenat and Pavel Shvaiko. *Ontology Matching*. Springer, Heidelberg (DE), 2007.
13. Alfio Ferrara, Stefano Montanelli, Jan Noessner, and Heiner Stuckenschmidt. Benchmarking matching applications on the semantic web. In *Proc. of the 8th Extended Semantic Web Conference (ESWC 2011)*, Heraklion, Greece, 2011.
14. Alfio Ferrara, Andriy Nikolov, and Francois Scharffe. Data linking for the semantic web. *International Journal on Semantic Web and Information Systems*, 7(3):46–76, 2011.
15. E. Jiménez-Ruiz, B. Cuenca Grau, and I. Horrocks. On the feasibility of using OWL 2 DL reasoners for ontology matching problems. In *OWL Reasoner Evaluation Workshop*, 2012.
16. Ernesto Jimenez-Ruiz and Bernardo Cuenca Grau. LogMap: Logic-based and scalable ontology matching. In *Int'l Sem. Web Conf. (ISWC)*, pages 273–288, 2011.
17. Ernesto Jiménez-Ruiz, Bernardo Cuenca Grau, Ian Horrocks, and Rafael Berlanga. Logic-based assessment of the compatibility of UMLS ontology sources. *J. Biomed. Sem.*, 2, 2011.
18. Philipp Mayr and Vivien Petras. Building a terminology network for search: The KoMoHe project. In *Proc. of the Int. Conference on Dublin Core and Metadata Applications*, pages 177 – 182, 2008.
19. Christian Meilicke. *Alignment Incoherence in Ontology Matching*. PhD thesis, University Mannheim, 2011.
20. Christian Meilicke. *Alignment Incoherence in Ontology Matching*. PhD thesis, University of Mannheim, 2011.
21. Christian Meilicke, Raúl García-Castro, Fred Freitas, Willem Robert van Hage, Elena Montiel-Ponsoda, Ryan Ribeiro de Azevedo, Heiner Stuckenschmidt, Ondřej Šváb Zamazal, Vojtěch Svátek, Andrei Tamilin, Cássia Trojahn, and Shenghui Wang. MultiFarm: A benchmark for multilingual ontology matching. *Web Semantics: Science, Services and Agents on the World Wide Web*, (0):–, 2012.

22. Christian Meilicke and Heiner Stuckenschmidt. Incoherence as a basis for measuring the quality of ontology mappings. In *Proc. 3rd International Workshop on Ontology Matching (OM) collocated with ISWC*, pages 1–12, Karlsruhe (Germany), 2008.
23. Boris Motik, Rob Shearer, and Ian Horrocks. Hypertableau reasoning for description logics. *J. Artif. Intell. Res.*, 36:165–228, 2009.
24. Joachim Neubert. Bringing the "thesaurus for economics" on to the web of linked data. In *Proc. of the WWW Workshop on Linked Data on the Web (LDOW)*, 2009.
25. Maria Roşoiu, Cássia Trojahn dos Santos, and Jérôme Euzenat. Ontology matching benchmarks: generation and evaluation. In Pavel Shvaiko, Isabel Cruz, Jérôme Euzenat, Tom Heath, Ming Mao, and Christoph Quix, editors, *Proc. 6th International Workshop on Ontology Matching (OM) collocated with ISWC, Bonn (Germany)*, 2011.
26. Pavel Shvaiko and Jérôme Euzenat. Ontology matching: state of the art and future challenges. *IEEE Transactions on Knowledge and Data Engineering*, 2012, to appear.
27. York Sure, Oscar Corcho, Jérôme Euzenat, and Todd Hughes, editors. *Proc. of the Workshop on Evaluation of Ontology-based Tools (EON) collocated with ISWC*, Hiroshima (Japan), 2004.
28. Cássia Trojahn dos Santos, Christian Meilicke, Jérôme Euzenat, and Heiner Stuckenschmidt. Automating OAEI campaigns (first report). In *Proc. International Workshop on Evaluation of Semantic Technologies (iWEST) collocated with ISWC*, Shanghai (China), 2010.
29. Michael Uschold and Michael Gruninger. Ontologies and semantics for seamless connectivity. *SIGMOD Rec.*, 33(4):58–64, 2004.
30. Shenghui Wang, Antoine Isaac, Stefan Schlobach, Lourens van der Meij, and Balthasar Schopman. Instance-based semantic interoperability in the cultural heritage. *Semantic Web*, 3(1):45–64, 2012.
31. Benjamin Zapilko, Johann Schaible, Philipp Mayr, and Brigitte Mathiak. TheSoz: A SKOS representation of the thesaurus for the social sciences. *Semantic Web – Interoperability, Usability, Applicability*, 2012. accepted.

Grenoble, Milano, Amsterdam, Mannheim, Milton-Keynes, Montpellier, Trento,
Prague, Toulouse, Köln, Oxford,
December 2012