Results of the Ontology Alignment Evaluation Initiative 2015*

Michelle Cheatham[1], Zlatan Dragisic[2], Jérôme Euzenat[3], Daniel Faria[4], Alfio Ferrara[5], Giorgos Flouris[6], Irini Fundulaki[6], Roger Granada[7], Valentina Ivanova[2], Ernesto Jiménez-Ruiz[8], Patrick Lambrix[2], Stefano Montanelli[5], Catia Pesquita[9], Tzanina Saveta[6], Pavel Shvaiko[10], Alessandro Solimando[11], Cássia Trojahn[7], and Ondřej Zamazal[12]

[1] Data Semantics (DaSe) Laboratory, Wright State University, USA
michelle.cheatham@wright.edu
[2] Linköping University & Swedish e-Science Research Center, Linköping, Sweden
{zlatan.dragisic,valentina.ivanova,patrick.lambrix}@liu.se
[3] INRIA & Univ. Grenoble Alpes, Grenoble, France
Jerome.Euzenat@inria.fr
[4] Instituto Gulbenkian de Ciência, Lisbon, Portugal
dfaria@igc.gulbenkian.pt
[5] Università degli studi di Milano, Italy
{alfio.ferrara,stefano.montanelli}@unimi.it
[6] Institute of Computer Science-FORTH, Heraklion, Greece
{jsaveta,fgeo,fundul}@ics.forth.gr
[7] IRIT & Université Toulouse II, Toulouse, France
{roger.granada,cassia.trojahn}@irit.fr
[8] University of Oxford, UK
ernesto@cs.ox.ac.uk
[9] LASIGE, Faculdade de Ciências, Universidade de Lisboa, Portugal
cpesquita@di.fc.ul.pt
[10] TasLab, Informatica Trentina, Trento, Italy
pavel.shvaiko@infotn.it
[11] INRIA-Saclay & Univ. Paris-Sud, Orsay, France
alessandro.solimando@inria.fr
[12] University of Economics, Prague, Czech Republic
ondrej.zamazal@vse.cz

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 and consensus. OAEI 2015 offered 8 tracks with 15 test cases followed by 22 participants. Since 2011, 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 2015 campaign.

* 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 [14, 17]. 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) [38]. 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) [2]. Starting from 2006 through 2014 the OAEI campaigns were held at the Ontology Matching workshops collocated with ISWC [15, 13, 4, 10–12, 1, 6, 9]. In 2015, the OAEI results were presented again at the Ontology Matching workshop[2] collocated with ISWC, in Bethlehem, PA US.

Since 2011, we have been using 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. For OAEI 2015, almost all of the OAEI data sets were evaluated under the SEALS modality, providing a more uniform evaluation setting. This year we did not continue the library track, however we significantly extended the evaluation concerning the conference, interactive and instance matching tracks. Furthermore, the multifarm track was extended with Arabic and Italian as languages.

This paper synthetizes the 2015 evaluation campaign and introduces the results provided in the papers of the participants. The remainder of the paper is organised 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 11 overviews lessons learned from the campaign. Finally, Section 12 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://om2015.ontologymatching.org
[3] http://www.seals-project.eu

2.1 Tracks and test cases

This year's campaign consisted of 8 tracks gathering 15 test cases 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, we generated a new benchmark based on the original bibliographic ontology and another benchmark using an energy ontology.

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

- **Anatomy (§4):** The anatomy test case is about matching the Adult Mouse Anatomy (2744 classes) and a small fragment of the NCI Thesaurus (3304 classes) describing the human anatomy.
- **Conference (§5):** The goal of the conference test case is to find all correct correspondences within a collection of ontologies describing the domain of organizing conferences. Results were evaluated automatically against reference alignments and by using logical reasoning techniques.
- **Large biomedical ontologies (§6):** The largebio test case aims at finding alignments between large and semantically rich biomedical ontologies such as FMA, SNOMED-CT, and NCI. The UMLS Metathesaurus has been used as the basis for reference alignments.

Multilingual

- **Multifarm (§7):** This test case is based on a subset of the Conference data set, translated into eight different languages (Chinese, Czech, Dutch, French, German, Portuguese, Russian, and Spanish) and the corresponding alignments between these ontologies. Results are evaluated against these alignments. This year, translations involving Arabic and Italian languages have been added.

Interactive matching

- **Interactive (§8):** This test case offers the possibility to compare different matching tools which can benefit from user interaction. Its goal is to show if user interaction can improve matching results, which methods are most promising and how many interactions are necessary. Participating systems are evaluated on the conference data set using an oracle based on the reference alignment.

Ontology Alignment For Query Answering OA4QA (§9): This test case offers the possibility to evaluate alignments in their ability to enable query answering in an ontology based data access scenario, where multiple aligned ontologies exist. In addition, the track is intended as a possibility to study the practical effects of logical violations affecting the alignments, and to compare the different repair strategies adopted by the ontology matching systems. In order to facilitate the understanding of the dataset and the queries, the conference data set is used, extended with synthetic ABoxes.

Instance matching (§10). The track is organized in five independent tasks and each task is articulated in two tests, namely *sandbox* and *mainbox*, with different scales, i.e., number of instances to match. The sandbox (small scale) is an open test, meaning that the set of expected mappings (i.e., reference alignment) is given in advance

test	formalism	relations	confidence	modalities	language	SEALS
benchmark	OWL	=	[0 1]	blind	EN	√
anatomy	OWL	=	[0 1]	open	EN	√
conference	OWL	=, <=	[0 1]	blind+open	EN	√
largebio	OWL	=	[0 1]	open	EN	√
multifarm	OWL	=	[0 1]	open+blind	AR, CZ, CN, DE, EN, ES, FR, IT, NL, RU, PT	√
interactive	OWL	=, <=	[0 1]	open	EN	√
OA4QA	OWL	=, <=	[0 1]	open	EN	
author-dis	OWL	=	[0 1]	open+blind	EN, IT	√
author-rec	OWL	=	[0 1]	open+blind	EN, IT	√
val-sem	OWL	<=	[0 1]	open+blind	EN	√
val-struct	OWL	<=	[0 1]	open+blind	EN	√
val-struct-sem	OWL	<=	[0 1]	open+blind	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).

to the participants. The mainbox (medium scale) is a blind test, meaning that the reference alignment is not given in advance to the participants. Each test contains two datasets called source and target and the goal is to discover the matching pairs, i.e., mappings or correspondences, among the instances in the source dataset and those in the target dataset.

Author-dis: The goal of the author-dis task is to link OWL instances referring to the same person (i.e., author) based on their publications.

Author-rec: The goal of the author-rec task is to associate a person, i.e., author, with the corresponding *publication report* containing aggregated information about the publication activity of the person, such as number of publications, h-index, years of activity, number of citations.

Val-sem: The goal of the val-sem task is to determine when two OWL instances describe the same Creative Work. The datasets of the val-sem task have been produced by altering a set of original data through value-based and semantics-aware transformations.

Val-struct: The goal of the val-struct task is to determine when two OWL instances describe the same Creative Work. The datasets of the val-struct task have been produced by altering a set of original data through value-based and structure-based transformations.

Val-struct-sem: The goal of the val-struct-sem task is to determine when two OWL instances describe the same Creative Work. The datasets of the val-struct-sem task have been produced by altering a set of original data through value-based, structure-based and semantics-aware transformations.

Table 1 summarizes the variation in the proposed test cases.

2.2 The SEALS platform

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. It 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 has to be uploaded on the SEALS portal[4]. Consequently, the evaluation can be executed by the organizers with the help of the SEALS infrastructure. This approach allowed to measure runtime and ensured the reproducibility of the results. As a side effect, this approach also ensures that a tool is executed with the same settings for all of the test cases that were executed in the SEALS mode.

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 3^{rd}, 2015. This gave 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 3^{rd}, 2015. The (open) 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 [8]. 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 3^{rd} and September 1^{st}, 2015.

2.5 Evaluation phase

Participants have been encouraged to upload their wrapped tools on the SEALS portal by September 1^{st}, 2015. 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, who had the occasion to fix their tools and resubmit them.

First results were available by October 1^{st}, 2015. The organizers provided these results individually to the participants. The results were published on the respective web pages by the organizers by October 15^{st}. The standard evaluation measures are usually precision and recall computed against the reference alignments. More details on evaluation measures are given in each test case section.

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

2.6 Comments on the execution

The number of participating systems has changed over the years with an increase tendency with some exceptional cases: 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, 23 in 2013, 14 in 2014. This year, we count on 22 systems. Furthermore participating systems are constantly changing, for example, this year 10 systems had not participated in any of the previous campaigns. The list of participants is summarized in Table 2. Note that some systems were also evaluated with different versions and configurations as requested by developers (see test case sections for details).

System	AML	CLONA	COMMAND	CroMatcher	DKP-AOM	DKP-AOM-Lite	EXONA	GMap	InsMT+	JarvisOM	Lily	LogMap	LogMapC	LogMap-Bio	LogMapLt	LYAM++	Mamba	RiMOM	RSDLWB	ServOMBI	STRIM	XMap	Total=22
Confidence	√	√	√	√				√			√	√	√	√		√		√		√		√	14
benchmarks	√	.		√	√			√			√	√	√		√		√		.	√		√	11
anatomy	√		√	√	√	√		√		√	√	√	√	√	√				√	√		√	15
conference	√		√	√	√			√		√	√	√	√		√		√		√	√		√	14
multifarm	√	√		√	√			√				√	√		√	√	√		√			√	12
interactive	√									√		√								√			4
largebio	√			√	√	√					√	√	√	√	√				√	√		√	12
OA4QA	√		√	√	√			√		√	√	√	√		√		√		√	√		√	14
instance							√		√		√	√						√			√		6
total	7	1	3	6	6	2	1	5	1	4	6	8	6	2	6	1	4	1	6	5	1	6	88

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

Finally, some systems were not able to pass some test cases as indicated in Table 2. The result summary per test case 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 by discarding and modifying features from a seed ontology. Considered features are names of entities, comments, the specialization

hierarchy, instances, properties and classes. This test focuses on the characterization of the behavior of the tools rather than having them compete on real-life problems. Full description of the systematic benchmark test set can be found on the OAEI web site.

Since OAEI 2011.5, the test sets are generated automatically by the test generator described in [16] from different seed ontologies. This year, we used two ontologies:

biblio The bibliography ontology used in the previous years which concerns bibliographic references and is inspired freely from BibTeX;

energy energyresource[5] is an ontology representing energy information for smart home systems developed at the Technische Universität Wien.

The characteristics of these ontologies are described in Table 3.

Test set	biblio	energy
classes+prop	33+64	523+110
instances	112	16
entities	209	723
triples	1332	9331

Table 3. Characteristics of the two seed ontologies used in benchmarks.

The initially generated tests from the IFC4 ontology which was provided to participants was found to be "somewhat erroneous" as the reference alignments contained only entities in the prime ontology namespace. We thus generated the energy data set. This test has also created problems to some systems, but we decided to keep it as an example, especially that some other systems have worked on it regularly with decent results. Hence, it may be useful for developers to understand why this is the case.

The energy data set was not available to participants when they submitted their systems. The tests were also blind for the organizers since we did not look into them before running the systems.

The reference alignments are still restricted to named classes and properties and use the "=" relation with confidence of 1.

3.2 Results

Contrary to previous years, we have not been able to evaluate the systems in a uniform setting. This is mostly due to relaxing the policy for systems which were not properly packaged under the SEALS interface so that they could be seamlessly evaluated. Systems required extra software installation and extra software licenses which rendered evaluation uneasy.

Another reason of this situation is the limited availability of evaluators for installing software for the purpose of evaluation.

[5] `https://www.auto.tuwien.ac.at/downloads/thinkhome/ontology/EnergyResourceOntology.owl`

It was actually the goal of the SEALS project to automate this evaluation so that the tool installation burden was put on tool developers and the evaluation burden on evaluators. This also reflects the idea that a good tool is a tool easy to install, so in which the user does not have many reasons to not using it.

As a consequence, systems have been evaluated in three different machine configurations:

- edna, AML2014, AML, CroMatcher, GMap, Lily, LogMap-C, LogMapLt, LogMap and XMap were run on a Debian Linux virtual machine configured with four processors and 8GB of RAM running under a Dell PowerEdge T610 with 2*Intel Xeon Quad Core 2.26GHz E5607 processors and 32GB of RAM, under Linux ProxMox 2 (Debian). All matchers where run under the SEALS client using Java 1.8 and a maximum heap size of 8GB.
- DKP-AOM, JarvisOM, RSDLWB and ServOMBI were run on a Debian Linux virtual machine configured with four processors and 20GB of RAM running under a Dell PowerEdge T610 with 2*Intel Xeon Quad Core 2.26GHz E5607 processors and 32GB of RAM, under Linux ProxMox 2 (Debian).
- Mamba was run under Ubuntu 14.04 on a Intel Core i7-3537U 2.00GHz×4 CPU with 8GB of RAM.

Under such conditions, we cannot compare systems on the basis of their speed. Reported figures are the average of 5 runs.

Participation From the 21 systems participating to OAEI this year, 14 systems were evaluated in this track. Several of these systems encountered problems: We encountered problems with one very slow matcher (LogMapBio) that has been eliminated from the pool of matchers. AML and ServOMBI had to be killed while they were unable to match the second run of the energy data set. No timeout was explicitly set. We did not investigate these problems.

Compliance Table 4 synthesizes the results obtained by matchers.

Globally results are far better on the biblio test than the energy one. This may be due either to system overfit to biblio or to the energy dataset being erroneous. However, 5 systems obtained best overall F-measure on the energy data set (this is comparable to the results obtained in 2014). It seems that run 1, 4 and 5 of energy generated ontologies found erroneous by some parsers (the matchers did not return any results), but some matchers where able to return relevant results. Curiously XMap did only work properly on tests 2 and 3.

Concerning F-measure results, all tested systems are above edna with LogMap-C been lower (we excluded LogMapIM which is definitely dedicated to instance matching only as well as JarvisOM and RSDLWD which outputed no useful results). Lily and CroMatcher achieve impressive 90% and 88% F-measure. Not only these systems achieve a high precision but a high recall of 83% as well. CroMatcher maintains its good results on energy (while Lily cannot cope with the test), however LogMapLt obtain the best F-measure (of 77%) on energy.

Matcher	biblio			energy		
	Prec.	F-m.	Rec.	Prec.	F-m.	Rec.
edna	.35(.58)	.41(.54)	.51(.50)	.50(.74)	.42(.49)	.15(.15)
AML2014	.92(.94)	.55(.55)	.39(.39	.98(.95)	.71(.69)	.23(.22)
AML	.99(.99)	.57(.56)	.40(.40)	1.0(.96)	.17(.16)	.04(.04)
CroMatcher	.94(.68)	.88(.62)	.82(.57)	.96(.76)	.68(.50)	.21(.16)
DKP-AOM	NaN	NaN	0.	.67	.59	.21
GMap	.93(.74)	.68(.53)	.53(.41)	.32(.42)	.11(.03)	.02(.02)
Lily	.97(.45)	.90(.40)	.83(.36)	NaN	NaN	0.
LogMap-C	.42(.41)	.41(.39)	.39(.37)	NaN	NaN	0.
LogMapLt	.43	.46	.50	.74	.77	.81
LogMap	.93(.91)	.55(.52)	.40(.37)	NaN	NaN	0.
Mamba	.78	.56	.44	.83	.25	.06
ServOMBI	NaN	NaN	0.	.94	.06	.01
XMap	1.0	.57	.40	1.0	.51	.22

Table 4. Aggregated benchmark results: Harmonic means of precision, F-measure and recall, along with their confidence-weighted values (*: uncompleted results).

Last year we noted that the F-measure was lower than the previous year (with a 89% from YAM++ and already a 88% from CroMatcher in 2013). This year this level is reached again.

Like last year, we can consider that we have high-precision matchers, AML and XMap, achieving near perfect to perfect precision on both tests.

Polarity We draw the triangle graphs for the biblio tests (Figure 1). It confirms that systems are more precision-oriented than ever: no balaced system is visible in the middle of the graph (only Mamba has a more balanced behavior).

3.3 Conclusions

This year, matcher performances have again reached their best level on biblio. However, relaxation of constraints made many systems fail during the tests. Running on newly generated tests has proved more difficult (but different systems fail on different tests). Systems are still very oriented towards precision at the expense of recall.

4 Anatomy

The anatomy test case confronts matchers with a specific type of ontologies from the biomedical domain. We focus on two fragments of biomedical ontologies which describe the human anatomy[6] and the anatomy of the mouse[7]. This data set has been used since 2007 with some improvements over the years.

[6] http://www.cancer.gov/cancertopics/cancerlibrary/terminologyresources/

[7] http://www.informatics.jax.org/searches/AMA_form.shtml

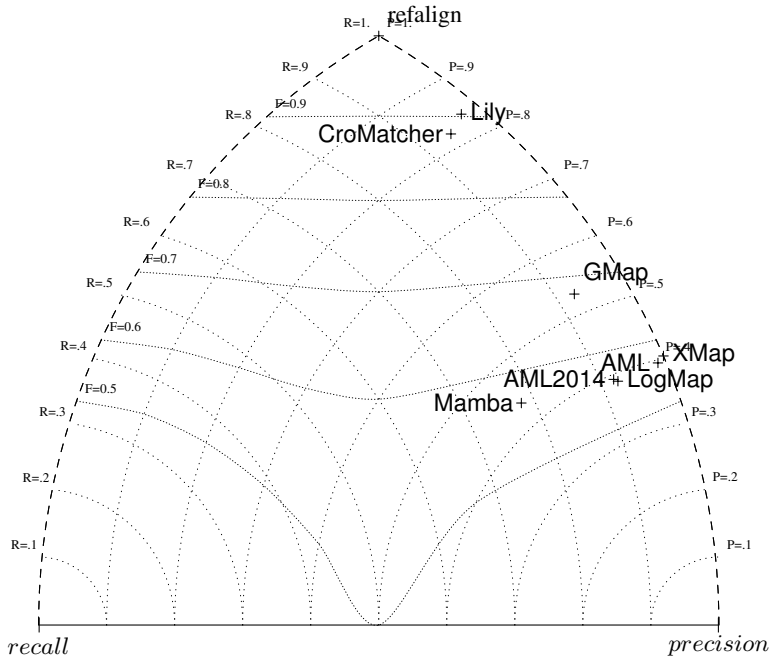

Fig. 1. Triangle view on the benchmark biblio data sets (run 5, non present systems have too low F-measure).

4.1 Experimental setting

We conducted experiments by executing each system in its standard setting and 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.

We run the systems on a server with 3.46 GHz (6 cores) and 8GB RAM allocated to each matching system. 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 removed trivial correspondences in the oboInOwl namespace like:

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

as well as correspondences expressing relations different from equivalence. Using the Pellet reasoner 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 5, we analyze all participating systems that could generate an alignment. The listing comprises 15 entries. LogMap participated with different versions, namely LogMap, LogMap-Bio, LogMap-C and a lightweight version LogMapLt that uses only some core components. Similarly, DKP-AOM is also participating with two versions, DKP-AOM and DKP-AOM-lite, DKP-AOM performs coherence analysis. There are systems which participate in the anatomy track for the first time. These are COMMAND, DKP-AOM, DKP-AOM-lite, GMap and JarvisOM. On the other hand, AML, LogMap (all versions), RSDLWB and XMap participated in the anatomy track last year while Lily and CroMatcher participated in 2011 and 2013 respectively. However, CroMatcher did not produce an alignment within the given timeframe in 2013. For more details, we refer the reader to the papers presenting the systems. Thus, this year we have 11 different systems (not counting different versions) which generated an alignment.

Three systems (COMMAND, GMap and Mamba) run out of memory and could not finish execution with the allocated amount of memory. Therefore, they were run on a different configuration with allocated 14 GB of RAM (Mamba additionally had database connection problems). Therefore, the execution times for COMMAND and GMap (marked with * and ** in the table) are not fully comparable to the other systems. As last year, we have 6 systems which finished their execution in less than 100 seconds. The top systems in terms of runtimes are LogMap, RDSLWB and AML. Depending on the specific version of the systems, they require between 20 and 40 seconds to match the ontologies. The table shows that there is no correlation between quality of the generated alignment in terms of precision and recall and required runtime. This result has also been observed in previous OAEI campaigns.

Table 5 also shows the results for precision, recall and F-measure. In terms of F-measure, the top ranked systems are AML, XMap, LogMap-Bio and LogMap. The results

Matcher	Runtime	Size	Precision	F-measure	Recall	Recall+	Coherent
AML	40	1477	0.96	0.94	0.93	0.82	√
XMap	50	1414	0.93	0.90	0.87	0.65	√
LogMapBio	895	1549	0.88	0.89	0.90	0.74	√
LogMap	24	1397	0.92	0.88	0.85	0.59	√
GMap	2362**	1344	0.92	0.86	0.81	0.53	-
CroMatcher	569	1350	0.91	0.86	0.81	0.51	-
Lily	266	1382	0.87	0.83	0.79	0.51	-
LogMapLt	20	1147	0.96	0.83	0.73	0.29	-
LogMap-C	49	1084	0.97	0.81	0.69	0.45	√
StringEquiv	-	946	1.00	0.77	0.62	0.00	-
DKP-AOM-lite	476	949	0.99	0.76	0.62	0.04	-
ServOMBI	792	971	0.96	0.75	0.62	0.10	-
RSDLWB	22	935	0.96	0.73	0.59	0.00	-
DKP-AOM	370	201	1.00	0.23	0.13	0.00	√
JarvisOM	217	458	0.37	0.17	0.11	0.01	-
COMMAND	63127*	150	0.29	0.05	0.03	0.04	√

Table 5. Comparison, ordered by F-measure, against the reference alignment, runtime is measured in seconds, the "size" column refers to the number of correspondences in the generated alignment.

of these four systems are at least as good as the results of the best systems in OAEI 2007-2010. AML, LogMap and LogMap-Bio produce very similar alignments compared to the last years. For example, AML's and LogMap's alignment contained only one correspondence less than the last year. Out of the systems which participated in the previous years, only Lily showed improvement. Lily's precision was improved from 0.81 to 0.87, recall from 0.73 to 0.79 and the F-measure from 0.77 to 0.83. This is also the first time that CroMatcher successfully produced an alignment given the set timeframe and its result is 6th best with respect to the F-measure.

This year we had 9 out of 15 systems which achieved an F-measure higher than the baseline which is based on (normalized) string equivalence (StringEquiv in the table). This is a slightly worse result (percentage-wise) than in the previous years when 7 out of 10 (2014) and 13 out of 17 systems (2012) produced alignments with F-measure higher than the baseline. The list of systems which achieved an F-measure lower than the baseline is comprised mostly of newly competing systems. The only exception is RSDLWB which competed last year when it also achieved a lower-than-baseline result.

Moreover, nearly all systems find many non-trivial correspondences. Exceptions are RSDLWB and DKP-AOM which generate only trivial correspondences.

This year seven systems produced coherent alignments which is comparable to the last year when 5 out of 10 systems achieved this.

4.3 Conclusions

This year we have again experienced an increase in the number of competing systems. The list of competing systems is comprised of both systems which participated in the previous years and new systems.

The evaluation of the systems has shown that most of the systems which participated in the previous years did not improve their results and in most cases they achieved slightly worse results. The only exception is Lily which showed some improvement compared to the previous time it competed. Out of the newly participating systems, GMap displayed the best performance and achieved the 5th best result with respect to the F-measure this year.

5 Conference

The conference test case requires matching several moderately expressive ontologies from the conference organization domain.

5.1 Test data

The data set consists of 16 ontologies in the domain of organizing conferences. These ontologies have been developed within the OntoFarm project[8].

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 and properties, in expressivity, but also in underlying resources.

5.2 Results

We provide results in terms of F-measure, comparison with baseline matchers and results from previous OAEI editions and precision/recall triangular graph based on sharp reference alignment. This year we newly provide results based on the uncertain version of reference alignment and on violations of consistency and conservativity principles.

[8] http://owl.vse.cz:8080/ontofarm/

Evaluation based on sharp reference alignments We evaluated the results of participants against blind reference alignments (labelled as *rar2*).[9] This includes all pairwise combinations between 7 different ontologies, i.e. 21 alignments.

These reference alignments have been made in two steps. First, we have generated them 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 by evaluators. The resulting reference alignments are labelled as *ra2*. Second, we detected violations of conservativity using the approach from [34] and resolved them by an evaluator. The resulting reference alignments are labelled as *rar2*. 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 approximations of the new ones.

Matcher	Prec.	$F_{0.5}$-m.	F_1-m.	F_2-m.	Rec.	Inc.Align.	Conser.V.	Consist.V.
AML	0.78	0.74	0.69	0.65	0.62	0	39	0
Mamba	0.78	0.74	0.68	0.64	0.61	2	85	16
LogMap-C	0.78	0.72	0.65	0.58	0.55	0	5	0
LogMap	0.75	0.71	0.65	0.6	0.57	0	29	0
XMAP	0.8	0.73	0.64	0.58	0.54	0	19	0
GMap	0.61	0.61	0.61	0.61	0.61	8	196	69
DKP-AOM	0.78	0.69	0.59	0.51	0.47	0	16	0
LogMapLt	0.68	0.62	0.56	0.5	0.47	3	97	18
edna	*0.74*	*0.66*	*0.56*	*0.49*	*0.45*			
ServOMBI	0.56	0.56	0.55	0.55	0.55	11	1325	235
COMMAND	0.72	0.64	0.55	0.48	0.44	14	505	235
StringEquiv	*0.76*	*0.65*	*0.53*	*0.45*	*0.41*			
CroMatcher	0.57	0.55	0.52	0.49	0.47	6	69	78
Lily	0.54	0.53	0.52	0.51	0.5	9	140	124
JarvisOM	0.8	0.64	0.5	0.4	0.36	2	27	7
RSDLWB	0.23	0.26	0.31	0.38	0.46	11	48	269

Table 6. The highest average $F_{[0.5|1|2]}$-measure and their corresponding precision and recall for each matcher with its F_1-optimal threshold (ordered by F_1-measure). Inc.Align. means number of incoherent alignments. Conser.V. means total number of all conservativity principle violations. Consist.V. means total number of all consistency principle violations.

Table 6 shows the results of all participants with regard to the reference alignment *rar2*. $F_{0.5}$-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

[9] More details about evaluation applying other sharp reference alignments are available at the conference web page.

$F_{0.5}$ weights precision higher than recall. The matchers shown in the table are ordered according to their highest average F_1-measure. We employed two baseline matchers. edna (string edit distance matcher) is used within the benchmark test case and with regard to performance it is very similar as the previously used *baseline2* in the conference track; StringEquiv is used within the anatomy test case. These baselines divide matchers into three performance groups. Group 1 consists of matchers (AML, Mamba, LogMap-C, LogMap, XMAP, GMap, DKP-AOM and LogMapLt) having better (or the same) results than both baselines in terms of highest average F_1-measure. Group 2 consists of matchers (ServOMBI and COMMAND) performing better than baseline StringEquiv. Other matchers (CroMatcher, Lily, JarvisOM and RSDLWB) performed slightly worse than both baselines. The performance of all matchers regarding their precision, recall and F_1-measure is visualized in Figure 2. Matchers are represented as squares or triangles. Baselines are represented as circles.

Further, we evaluated performance of matchers separately on classes and properties. We compared position of tools within overall performance groups and within only class performance groups. We observed that on the one side ServOMBI and LogMapLt improved their position in overall performance groups wrt. their position in only classes performance groups due to their better property matching performance than baseline edna. On the other side RSDLWB worsen its position in overall performance groups wrt. its position in only classes performance groups due to its worse property matching performance than baseline StringEquiv. DKP-AOM and Lily do not match properties at all but they remained in their respective overall performance groups wrt. their positions in only classes performance groups. More details about these evaluation modalities are on the conference web page.

Comparison with previous years wrt. ra2 Six matchers also participated in this test case in OAEI 2014. The largest improvement was achieved by XMAP (recall from .44 to .51, while precision decreased from .82 to .81), and AML (precision from .80 to .81 and recall from .58 to .61). Since we applied *rar2* reference alignment for the first time, we used *ra2*, consistent but not conservativity violations free, reference alignment for year-by-year comparison.

Evaluation based on uncertain version of reference alignments The confidence values of all correspondences in the sharp reference alignments for the conference track are all 1.0. For the uncertain version of this track, the confidence value of a correspondence has been set equal to the percentage of a group of people who agreed with the correspondence in question (this uncertain version is based on reference alignment labelled as *ra1*). One key thing to note is that the group was only asked to validate correspondences that were already present in the existing reference alignments – so some correspondences had their confidence value reduced from 1.0 to a number near 0, but no new correspondence was added.

There are two ways that we can evaluate matchers according to these "uncertain" reference alignments, which we refer to as *discrete* and *continuous*. The discrete evaluation considers any correspondence in the reference alignment with a confidence value of 0.5 or greater to be fully correct and those with a confidence less than 0.5 to be fully

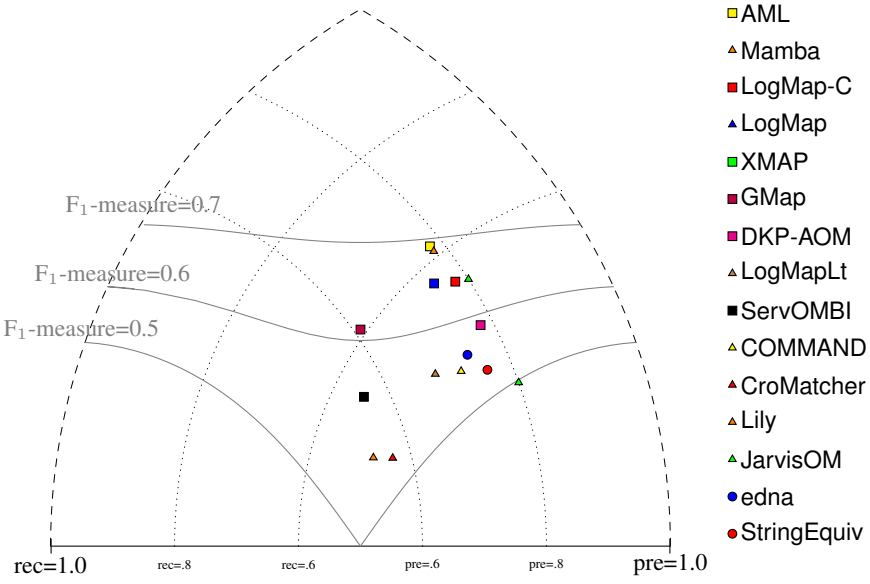

Fig. 2. Precision/recall triangular graph for the conference test case wrt. the *rar2* reference alignment. Dotted lines depict level of precision/recall while values of F_1-measure are depicted by areas bordered by corresponding lines F_1-measure=0.[5|6|7].

incorrect. Similarly, an matcher's correspondence is considered a "yes" if the confidence value is greater than or equal to the matcher's threshold and a "no" otherwise. In essence, this is the same as the "sharp" evaluation approach, except that some correspondences have been removed because less than half of the crowdsourcing group agreed with them. The continuous evaluation strategy penalizes an alignment system more if it misses a correspondence on which most people agree than if it misses a more controversial correspondence. For instance, if $A \equiv B$ with a confidence of 0.85 in the reference alignment and a matcher gives that correspondence a confidence of 0.40, then that is counted as $0.85 \times 0.40 = 0.34$ of a true positive and $0.85–0.40 = 0.45$ of a false negative.

	Sharp			Discrete			Continuous		
Matcher	Prec.	F_1-m.	Rec.	Prec.	F_1-m.	Rec.	Prec.	F_1-m.	Rec.
AML	0.84	0.74	0.66	0.82	0.72	0.65	0.8	0.76	0.73
COMMAND	0.78	0.59	0.47	0.76	0.61	0.51	0.6	0.53	0.47
CroMatcher	0.59	0.54	0.5	0.57	0.55	0.53	0.58	0.51	0.46
DKP-AOM	0.84	0.63	0.5	0.83	0.62	0.5	0.8	0.69	0.61
GMap	0.66	0.65	0.65	0.65	0.64	0.64	0.63	0.61	0.58
JarvisOM	0.84	0.51	0.37	0.83	0.51	0.37	0.83	0.6	0.46
Lily	0.59	0.56	0.53	0.58	0.56	0.54	0.58	0.32	0.22
LogMap	0.8	0.68	0.59	0.78	0.68	0.6	0.76	0.63	0.54
LogMap-C	0.82	0.67	0.57	0.8	0.67	0.58	0.79	0.63	0.53
LogMapLt	0.73	0.59	0.5	0.72	0.58	0.49	0.71	0.66	0.62
Mamba	0.83	0.72	0.64	0.82	0.71	0.63	0.76	0.75	0.74
RSDLWB	0.25	0.33	0.49	0.23	0.32	0.51	0.23	0.33	0.64
ServOMBI	0.61	0.59	0.58	0.59	0.57	0.55	0.56	0.61	0.66
XMap	0.85	0.68	0.56	0.84	0.67	0.56	0.81	0.73	0.66

Table 7. F-measure, precision, and recall of the different matchers when evaluated using the sharp (s), discrete uncertain (d) and continuous uncertain (c) metrics.

The results from this year, see Table 7, follow the same general pattern as the results from the 2013 systems discussed in [5]. Out of the 14 matchers, five (DKP-AOM, JarvisOm, LogMapLt, Mamba, and RSDLWB) use 1.0 as the confidence values for all correspondences they identify. Two (ServOMBI and XMap) of the remaining nine have some variation in confidence values, though the majority are 1.0. The rest of the matchers have a fairly wide variation of confidence values. Most of these are near the upper end of the [0,1] range. The exception is Lily, which produces many correspondences with confidence values around 0.5.

Discussion In most cases, precision using the uncertain version of the reference alignment is the same or less than in the sharp version, while recall is slightly greater with the uncertain version. This is because no new correspondence was added to the reference alignments, but controversial ones were removed.

Regarding differences between the discrete and continuous evaluations using the uncertain reference alignments, they are in general quite small for precision. This is because of the fairly high confidence values assigned by the matchers. COMMAND's continuous precision is much lower because it assigns very low confidence values to some correspondences in which the labels are equivalent strings, which many crowdsourcers agreed with unless there was a compelling contextual reason not to. Applying a low threshold value (0.53) for the matcher hides this issue in the discrete case, but the continuous evaluation metrics do not use a threshold.

Recall measures vary more widely between the discrete and continuous metrics. In particular, matchers that set all confidence values to 1.0 see the biggest gains between the discrete and continuous recall on the uncertain version of the reference alignment. This is because in the discrete case incorrect correspondences produced by those systems are counted as a whole false positive, whereas in the continuous version, they are penalized a fraction of that if not many people agreed with the correspondence. While this is interesting in itself, this is a one-time gain in improvement. Improvement on this metric from year-to-year will only be possible if developers modify their systems to produce meaningful confidence values. Another thing to note is the large drop in Lily's recall between the discrete and continuous approaches. This is because the confidence values assigned by that alignment system are in a somewhat narrow range and universally low, which apparently does not correspond well to human evaluation of the correspondence quality.

Evaluation based on violations of consistency and conservativity principles This year we performed evaluation based on detection of conservativity and consistency violations [34]. The consistency principle states that correspondences should not lead to unsatisfiable classes in the merged ontology; the conservativity principle states that correspondences should not introduce new semantic relationships between concepts from one of the input ontologies.

Table 6 summarizes statistics per matcher. There are ontologies that have unsatisfiable TBox after ontology merge (Uns.Ont.), total number of all conservativity principle violations within all alignments (Conser.V.) and total number of all consistency principle violations (Consist.V.).

Five tools (AML, DKP-AOM, LogMap, LogMap-C and XMAP) do not violate consistency. The lowest number of conservativity violations was achieved by LogMap-C which has a repair technique for them. Four further tools have an average of conservativity principle around 1 (DKP-AOM, JarvisOM, LogMap and AML).[10] We should note that these conservativity principle violations can be "false positives" since the entailment in the aligned ontology can be correct although it was not derivable in the single input ontologies.

In conclusion, this year eight matchers (against five matchers last year for easier reference alignment) performed better than both baselines on new, not only consistent but also conservative, reference alignments. Next two matchers perform almost equally well as the best baseline. Further, this year five matchers generate coherent alignments (against four matchers last year). Based on uncertain reference alignments many more

[10] All matchers but one delivered all 21 alignments. RSDLWB generated 18 alignments.

matchers provide alignments with a range of confidence values than in the past. This evaluation modality will enable us to evaluate degree of convergence between this year's results and humans scores on the alignment task next years.

6 Large biomedical ontologies (largebio)

The largebio test case 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.

6.1 Test data

The test case has been split into three matching problems: FMA-NCI, FMA-SNOMED and SNOMED-NCI; and each matching problem in 2 tasks involving different fragments of the input ontologies.

The UMLS Metathesaurus [3] has been selected as the basis for 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 sense of [17]) between the integrated ontologies, it is relatively straightforward to extract them from the information provided in the distribution files (see [21] 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 [21].

Since alignment coherence is an aspect of ontology matching that we aim to promote, in previous editions we provided coherent reference alignments by refining the UMLS mappings using the Alcomo (alignment) debugging system [26], LogMap's (alignment) repair facility [20], or both [22].

However, concerns were raised about the validity and fairness of applying automated alignment repair techniques to make reference alignments coherent [30]. It is clear that using the original (incoherent) UMLS alignments would be penalizing to ontology matching systems that perform alignment repair. However, using automatically repaired alignments would penalize systems that do not perform alignment repair and also systems that employ a repair strategy that differs from that used on the reference alignments [30].

Thus, as in the 2014 edition, we arrived at a compromising solution that should be fair to all ontology matching systems. Instead of repairing the reference alignments as normal, by removing correspondences, we flagged the *incoherence-causing correspondences* in the alignments by setting the relation to *"?"* (unknown). These "?" correspondences will neither be considered as positive nor as negative when evaluating the participating ontology matching systems, but will simply be ignored. This way, systems that do not perform alignment repair are not penalized for finding correspondences that (despite causing incoherences) may or may not be correct, and systems that do perform alignment repair are not penalized for removing such correspondences.

To ensure that this solution was as fair as possible to all alignment repair strategies, we flagged as unknown all correspondences suppressed by any of Alcomo, LogMap or AML [31], as well as all correspondences suppressed from the reference alignments of last year's edition (using Alcomo and LogMap combined). Note that, we have used the (incomplete) repair modules of the above mentioned systems.

The flagged UMLS-based reference alignment for the OAEI 2015 campaign is summarized in Table 8.

Table 8. Respective sizes of reference alignments

Reference alignment	"=" corresp.	"?" corresp.
FMA-NCI	2,686	338
FMA-SNOMED	6,026	2,982
SNOMED-NCI	17,210	1,634

6.2 Evaluation setting, participation and success

We have run the evaluation in a Ubuntu Laptop with an Intel Core i7-4600U CPU @ 2.10GHz x 4 and allocating 15Gb of RAM. Precision, Recall and F-measure have been computed with respect to the UMLS-based reference alignment. Systems have been ordered in terms of F-measure.

In the OAEI 2015 largebio track, 13 out of 22 participating OAEI 2015 systems have been able to cope with at least one of the tasks of the largebio track. Note that RiMOM-IM, InsMT+, STRIM, EXONA, CLONA and LYAM++ are systems focusing on either the instance matching track or the multifarm track, and they did not produce any alignment for the largebio track. COMMAND and Mamba did not finish the smallest largebio task within the given 12 hours timeout, while GMap and JarvisOM gave an "error exception" when dealing with the smallest largebio task.

6.3 Background knowledge

Regarding the use of background knowledge, LogMap-Bio uses BioPortal as mediating ontology provider, that is, it retrieves from BioPortal the most suitable top-10 ontologies for the matching task.

LogMap uses normalisations and spelling variants from the general (biomedical) purpose UMLS Lexicon.

AML has three sources of background knowledge which can be used as mediators between the input ontologies: the Uber Anatomy Ontology (Uberon), the Human Disease Ontology (DOID) and the Medical Subject Headings (MeSH).

XMAP has been evaluated with two variants: XMAP-BK and XMAP. XMAP-BK uses synonyms provided by the UMLS Metathesaurus, while XMAP has this feature deactivated. **Note that matching systems using UMLS-Metathesaurus as background**

System	FMA-NCI		FMA-SNOMED		SNOMED-NCI		Average	#
	Task 1	Task 2	Task 3	Task 4	Task 5	Task 6		
LogMapLt	16	213	36	419	212	427	221	6
RSDLWB	17	211	36	413	221	436	222	6
AML	36	262	79	509	470	584	323	6
XMAP	26	302	46	698	394	905	395	6
XMAP-BK	31	337	49	782	396	925	420	6
LogMap	25	265	78	768	410	1,062	435	6
LogMapC	106	569	156	1,195	3,039	3,553	1,436	6
LogMapBio	1,053	1,581	1,204	3,248	3,298	3,327	2,285	6
ServOMBI	234	-	532	-	-	-	383	2
CroMatcher	2,248	-	13,057	-	-	-	7,653	2
Lily	740	-	-	-	-	-	740	1
DKP-AOM	1,491	-	-	-	-	-	1,491	1
DKP-AOM-Lite	1,579	-	-	-	-	-	1,579	1
# Systems	**13**	**10**	**8**	**8**	**8**	**8**	**1,353**	**55**

Table 9. System runtimes (s) and task completion.

knowledge will have a *notable advantag*e since the largebio reference alignment is also based on the UMLS-Metathesaurus. Nevertheless, it is still interesting to evaluate the performance of a system with and without the use of the UMLS-Metathesaurus.

6.4 Alignment coherence

Together with Precision, Recall, F-measure and Runtimes we have also evaluated the coherence of alignments. We report (1) the number of unsatisfiabilities when reasoning with the input ontologies together with the computed alignments, and (2) the ratio of unsatisfiable classes with respect to the size of the union of the input ontologies.

We have used the OWL 2 reasoner HermiT [28] to compute the number of unsatisfiable classes. For the cases in which MORe could not cope with the input ontologies and the alignments (in less than 2 hours) we have provided a lower bound on the number of unsatisfiable classes (indicated by $\geq$) using the OWL 2 EL reasoner ELK [23].

In this OAEI edition, only two systems have shown alignment repair facilities, namely: AML and LogMap (including LogMap-Bio and LogMap-C variants). Tables 10-13 (see last two columns) show 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 if they are to be used in tasks involving reasoning.

6.5 Runtimes and task completion

Table 9 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 the average time required to complete them. Times are reported in seconds.

Task 1: small FMA and NCI fragments							
System	Time (s)	# Corresp.	Scores			Incoherence	
			Prec.	F-m.	Rec.	Unsat.	Degree
XMAP-BK *	31	2,714	0.97	0.93	0.90	2,319	22.6%
AML	36	2,690	0.96	0.93	0.90	2	0.019%
LogMap	25	2,747	0.95	0.92	0.90	2	0.019%
LogMapBio	1,053	2,866	0.93	0.92	0.92	2	0.019%
LogMapLt	16	2,483	0.97	0.89	0.82	2,045	19.9%
ServOMBI	234	2,420	0.97	0.88	0.81	3,216	31.3%
XMAP	26	2,376	0.97	0.87	0.78	2,219	21.6%
LogMapC	106	2,110	0.96	0.82	0.71	2	0.019%
Average	584	2,516	0.85	0.78	0.73	2,497	24.3%
Lily	740	3,374	0.60	0.66	0.72	9,279	90.2%
DKP-AOM-Lite	1,579	2,665	0.64	0.62	0.60	2,139	20.8%
DKP-AOM	1,491	2,501	0.65	0.61	0.57	1,921	18.7%
CroMatcher	2,248	2,806	0.57	0.57	0.57	9,301	90.3%
RSDLWB	17	961	0.96	0.48	0.32	25	0.2%

Task 2: whole FMA and NCI ontologies							
System	Time (s)	# Corresp.	Scores			Incoherence	
			Prec.	F-m.	Rec.	Unsat.	Degree
XMAP-BK *	337	2,802	0.87	0.86	0.85	1,222	0.8%
AML	262	2,931	0.83	0.84	0.86	10	0.007%
LogMap	265	2,693	0.85	0.83	0.80	9	0.006%
LogMapBio	1,581	3,127	0.77	0.81	0.85	9	0.006%
XMAP	302	2,478	0.87	0.80	0.74	1,124	0.8%
Average	467	2,588	0.82	0.76	0.73	3,742	2.6%
LogMapC	569	2,108	0.88	0.75	0.65	9	0.006%
LogMapLt	213	3,477	0.67	0.74	0.82	26,478	18.1%
RSDLWB	211	1,094	0.80	0.44	0.31	1,082	0.7%

Table 10. Results for the FMA-NCI matching problem. * Uses background knowledge based on the UMLS-Metathesaurus as the largebio reference alignments.

The last column reports the number of tasks that a system could complete. For example, 8 system were able to complete all six tasks. The last row shows the number of systems that could finish each of the tasks. The tasks involving SNOMED were also harder with respect to both computation times and the number of systems that completed the tasks.

6.6 Results for the FMA-NCI matching problem

Table 10 summarizes the results for the tasks in the FMA-NCI matching problem. The following tables summarize the results for the tasks in the FMA-NCI matching problem.

XMAP-BK and AML provided the best results in terms of F-measure in Task 1 and Task 2. Note that, the use of background knowledge based on the UML-Metathesaurus

Task 3: small FMA and SNOMED fragments							
System	Time (s)	# Corresp.	Scores			Incoherence	
			Prec.	F-m.	Rec.	Unsat.	Degree
XMAP-BK *	49	7,920	0.97	0.90	0.85	12,848	54.4%
AML	79	6,791	0.93	0.82	0.74	0	0.0%
LogMapBio	1,204	6,485	0.94	0.80	0.70	1	0.004%
LogMap	78	6,282	0.95	0.80	0.69	1	0.004%
ServOMBI	532	6,329	0.96	0.79	0.66	12,155	51.5%
XMAP	46	6,133	0.96	0.77	0.65	12,368	52.4%
Average	1,527	5,328	0.92	0.66	0.56	5,902	25.0%
LogMapC	156	4,535	0.96	0.66	0.51	0	0.0%
CroMatcher	13,057	6,232	0.59	0.53	0.48	20,609	87.1%
LogMapLt	36	1,644	0.97	0.34	0.21	771	3.3%
RSDLWB	36	933	0.98	0.23	0.13	271	1.1%

Task 4: whole FMA ontology with SNOMED large fragment							
System	Time (s)	# Corresp.	Scores			Incoherence	
			Prec.	F-m.	Rec.	Unsat.	Degree
XMAP-BK *	782	9,243	0.77	0.80	0.84	44,019	21.8%
AML	509	6,228	0.89	0.75	0.65	0	0.0%
LogMap	768	6,281	0.84	0.72	0.63	0	0.0%
LogMapBio	3,248	6,869	0.78	0.71	0.65	0	0.0%
XMAP	698	7,061	0.72	0.66	0.61	40,056	19.9%
LogMapC	1,195	4,693	0.85	0.61	0.48	98	0.049%
Average	1,004	5,395	0.83	0.60	0.53	11,157	5.5%
LogMapLt	419	1,822	0.85	0.34	0.21	4,389	2.2%
RSDLWB	413	968	0.93	0.22	0.13	698	0.3%

Table 11. Results for the FMA-SNOMED matching problem. * Uses background knowledge based on the UMLS-Metathesaurus as the largebio reference alignments.

has an important impact in the performance of XMAP-BK. LogMap-Bio improves LogMap's recall in both tasks, however precision is damaged specially in Task 2.

Note that efficiency in Task 2 has decreased 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. Furthermore, ServOMBI, CroMacther, LiLy, DKP-AOM-Lite and DKP-AOM could not complete Task 2.

6.7 Results for the FMA-SNOMED matching problem

Table 11 summarizes the results for the tasks in the FMA-SNOMED matching problem. XMAP-BK provided the best results in terms of both Recall and F-measure in Task 3 and Task 4. Precision of XMAP-BK in Task 2 was lower than the other top systems but Recall was much higher than the others.

Task 5: small SNOMED and NCI fragments							
System	Time (s)	# Corresp.	Scores			Incoherence	
			Prec.	F-m.	Rec.	Unsat.	Degree
AML	470	14,141	0.92	0.81	0.72	≥0	≥0.0%
LogMapBio	3,298	12,855	0.94	0.79	0.67	≥0	≥0.0%
LogMap	410	12,384	0.96	0.78	0.66	≥0	≥0.0%
XMAP-BK *	396	11,674	0.93	0.73	0.61	≥1	≥0.001%
XMAP	394	11,674	0.93	0.73	0.61	≥1	≥0.001%
LogMapLt	212	10,942	0.95	0.71	0.57	≥60,450	≥80.4%
Average	1,055	11,092	0.94	0.70	0.58	12,262	16.3%
LogMapC	3,039	9,975	0.91	0.65	0.51	≥0	≥0.0%
RSDLWB	221	5,096	0.97	0.42	0.27	≥37,647	≥50.0%

Task 6: whole NCI ontology with SNOMED large fragment							
System	Time (s)	# Corresp.	Scores			Incoherence	
			Prec.	F-m.	Rec.	Unsat.	Degree
AML	584	12,821	0.90	0.76	0.65	≥2	≥0.001%
LogMapBio	3,327	12,745	0.85	0.71	0.61	≥4	≥0.002%
LogMap	1,062	12,222	0.87	0.71	0.60	≥4	≥0.002%
XMAP-BK *	925	10,454	0.91	0.68	0.54	≥0	≥0.0%
XMAP	905	10,454	0.91	0.67	0.54	≥0	≥0.0%
LogMapLt	427	12,894	0.80	0.66	0.57	≥150,656	≥79.5%
Average	1,402	10,764	0.88	0.65	0.53	29,971	15.8%
LogMapC	3,553	9,100	0.88	0.60	0.45	≥2	≥0.001%
RSDLWB	436	5,427	0.89	0.41	0.26	≥89,106	≥47.0%

Table 12. Results for the SNOMED-NCI matching problem. * Uses background knowledge based on the UMLS-Metathesaurus as the largebio reference alignments.

As in the FMA-NCI tasks, the use of the UMLS-Metathesaurus in XMAP-BK has an important impact. Overall, the results were less positive than in 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 LogMapBio and XMAP in terms of precision. Furthermore, LiLy, DKP-AOM-Lite and DKP-AOM could not complete neither Task 3 nor Task 4, while ServOMBI and CroMatcher could not complete Task 4 within the permitted time.

6.8 Results for the SNOMED-NCI matching problem

Table 12 summarizes the results for the tasks in the SNOMED-NCI matching problem. AML provided the best results in terms of both Recall and F-measure in Task 5 and 6, while RSDLWB and XMAP provided the best results in terms of precision in Task 5 and 6, respectively.

Unlike in the FMA-NCI and FMA-SNOMED matching problems, the use of the UML-Metathesaurus did not impact the performance of XMAP-BK, which obtained almost identical results as XMAP. As in the previous matching problems, efficiency decreases as the ontology size increases. Furthermore, LiLy, DKP-AOM-Lite, DKP-AOM,

System	Total Time (s)	Average			
		Prec.	F-m.	Rec.	Inc. Degree
AML	1,940	0.90	0.82	0.75	0.005%
XMAP-BK *	2,520	0.90	0.82	0.76	16.6%
LogMap	2,608	0.90	0.79	0.71	0.005%
LogMapBio	13,711	0.87	0.79	0.73	0.005%
XMAP	2,371	0.89	0.75	0.65	15.8%
LogMapC	8,618	0.91	0.68	0.55	0.013%
LogMapLt	1,323	0.87	0.61	0.53	33.9%
RSDLWB	1,334	0.92	0.37	0.24	16.6%

Table 13. Summary results for the top systems. * Uses background knowledge based on the UMLS-Metathesaurus as the largebio reference alignments.

ServOMBI and CroMatcher could not complete neither Task 5 nor Task 6 in less than 12 hours.

6.9 Summary results for the top systems

Table 13 summarizes the results for the systems that completed all 6 tasks of largebio track. The table shows the total time in seconds to complete all tasks and averages for Precision, Recall, F-measure and Incoherence degree. The systems have been ordered according to the average F-measure and Incoherence degree.

AML and XMAP-BK were a step ahead and obtained the best average Recall and F-measure.

RSDLWB and LogMapC were the best systems in terms of precision.

Regarding incoherence, AML and LogMap variants (excluding LogMapLt) compute sets of correspondences leading to very small number of unsatisfiable classes.

Finally, LogMapLt and RSDLWB were the fastest system. Total computation times were slightly higher this year than previous years due to the (extra) overload of downloading the ontologies from the new SEALS repository.

6.10 Conclusions

Although the proposed matching tasks represent a significant leap in complexity with respect to the other OAEI test cases, the results have been very promising and 8 systems completed all matching tasks with very competitive results. Furthermore, 13 systems completed at least one of the tasks.

There is, as in previous OAEI campaigns, plenty of room for improvement: (1) most of the participating systems disregard the coherence of the generated alignments; (2) many system should improve scalability, , and (3) recall in the tasks involving SNOMED should be improved while keeping precision values.

The alignment coherence measure was the weakest point of the systems participating in this test case. As shown in Tables 10-13, even highly precise alignment sets may lead to a huge number of unsatisfiable classes (e.g. LogMapLt and RSDLWB alignments

in Task 5). The use of techniques to assess alignment coherence is critical if the input ontologies together with the computed alignments are to be used in practice. Unfortunately, only a few systems in OAEI 2015 have successfully used such techniques. We encourage ontology matching system developers to develop their own repair techniques or to use state-of-the-art techniques such as Alcomo [26], the repair module of LogMap (LogMap-Repair) [20] or the repair module of AML [31], which have worked well in practice [22, 18].

7 MultiFarm

The MultiFarm data set [27] aims at evaluating the ability of matching systems to deal with ontologies in different natural languages. This data set results from the translation of 7 ontologies from the conference track (cmt, conference, confOf, iasted, sigkdd, ekaw and edas), into 8 languages: Chinese, Czech, Dutch, French, German, Portuguese, Russian, and Spanish. For this campaign, Arabic and Italian translations have been also provided. With these two new languages, the data set is composed of 55 pairs of languages (see [27] for details on how the original MultiFarm data set has been generated). For each pair, taking into account the alignment direction (cmt$_{en}$–confOf$_{de}$ and cmt$_{de}$–confOf$_{en}$, for instance, as two distinct matching tasks), we have 49 matching tasks. The whole data set is composed of 55×49 matching tasks.

7.1 Experimental setting

Since 2014, part of the data set is used for blind evaluation. This subset includes all matching tasks involving the edas and ekaw ontologies (resulting in 55×24 matching tasks), which were not used in previous campaigns. In the rest of this paper, we refer to this blind evaluation as *edas and ekaw based evaluation*. Participants were able to test their systems on the available subset of matching tasks (*open evaluation*), available via the SEALS repository. The open subset covers 45×25 tasks[11].

We distinguish two types of matching tasks: *(i)* those tasks where two different ontologies (cmt–confOf, for instance) have been translated into two different languages; and *(ii)* those tasks where the same ontology (cmt–cmt) has been translated into two different languages. For the tasks of type (ii), good results are not directly related to the use of specific techniques for dealing with cross-lingual ontologies, but on the ability to exploit the identical structure of the ontologies.

In this campaign, 5 systems (out of 22 participants, see Table 2) implement cross-lingual matching strategies: AML, CLONA, LogMap, LYAM++ and XMap. This number increased with respect to the last campaign (3 in 2014). Most of them integrate a translation module in their implementations. LogMap uses Google Translator API and Microsoft Translation and pre-compiles a local dictionary in order to avoid multiple accesses to the translators within the matching process. AML, CLONA and XMap use Microsoft Translator, and AML and XMap adopt the same strategy of LogMap computing a

[11] This year, Italian translations have been only used in the blind setting.

local dictionary. All of them use English as pivot language. The translation step is performed before the matching step itself. An alternative strategy is adopted by LYAM++ which uses the multilingual resource BabelNet.

7.2 Execution setting and runtime

The systems have been executed on a Debian Linux VM configured with four processors and 20GB of RAM running under a Dell PowerEdge T610 with 2*Intel Xeon Quad Core 2.26GHz E5607 processors. The runtimes for both settings are shown in Tables 14 and 15. All measurements are based on a single run. Systems not listed in these tables were not wrapped using SEALS (COMMAND), are designed to deal with specific matching tasks (EXONA, InsMT, JarvisOM, RiMOM and ServOMBI), or generated empty alignments for all matching tasks (Lily).

For several reasons, some systems have been executed in a different setting (Mamba due to the issues with the Gurobi optimizer, LogMap due to network problems for accessing the translators, and LYAM++[12] due to issues with the BabelNet license). Thus, we do not report on execution time for these systems.

We can observe large differences between the time required for a system to complete the 45×25 (Table 14) and 55×24 (Table 15) matching tasks. However, we have experimented some problems when accessing the SEALS test repositories due to the many accesses to the server, i.e., tracks running their evaluations in parallel. Hence, the reported runtime may not reflect the real execution runtime required for completing the tasks.

7.3 Evaluation results

Open evaluation results. Table 14 presents the aggregated results for the open subset, for the test cases of type (i) and (ii)[13]. We do not apply any threshold on the confidence measure.

We observe significant differences between the results obtained for each type of matching task, specially in terms of precision, for most systems, with lower differences in terms of recall. As expected, in terms of F-measure, systems implementing cross-lingual techniques outperform the non-cross-lingual systems for test cases of type (i). For these cases, non-specific matchers have good precision but generating very few correspondences. While LogMap has the best precision (at the expense of recall), AML has similar results in terms of precision and recall and outperforms the other systems in terms of F-measure (this is the case for both types of tasks). For type (ii), CroMatcher takes advantage of the ontology structure and performs better than some specific cross-lingual systems.

With respect to the pairs of languages for test cases of type (i), for the sake of brevity, we do not present them here. The reader can refer to the OAEI results web page for detailed results for each of the 45 pairs. With exception of CroMatcher and RSDLWB,

[12] Exceptionally, for the open test, the alignments from LYAM++ have been provided by the developers instead of being generated under the SEALS platform.

[13] The results have been computed using the Alignment API 4.6.

			Type (i) – 20 tests per pair				Type (ii) – 5 tests per pair			
System	Time	#pairs	Size	Prec.	F-m.	Rec.	Size	Prec.	F-m.	Rec.
AML	10	45	11.58	.53$_{(.53)}$	**.51**$_{(.51)}$	**.50**$_{(.50)}$	58.29	.93$_{(.93)}$	**.64**$_{(.64)}$	**.50**$_{(.50)}$
CLONA	1629	45	9.45	.46$_{(.46)}$	.39$_{(.39)}$	.35$_{(.35)}$	50.89	.91$_{(.91)}$	.58$_{(.58)}$	.42$_{(.42)}$
LogMap*	36	45	6.37	**.75**$_{(.75)}$	.41$_{(.41)}$	.29$_{(.29)}$	42.83	**.95**$_{(.95)}$	.45$_{(.45)}$	.30$_{(.30)}$
LYAM++*	-	13	12.29	.14$_{(.50)}$	.14$_{(.49)}$	.14$_{(.44)}$	64.20	.26$_{(.90)}$	.19$_{(.66)}$	.15$_{(.53)}$
XMap	4012	45	36.39	.22$_{(.23)}$	.24$_{(.25)}$	.27$_{(.28)}$	61.65	.66$_{(.69)}$	.37$_{(.39)}$	.27$_{(.29)}$
CroMatcher	257	45	10.72	.30$_{(.30)}$	**.07**$_{(.07)}$	**.04**$_{(.04)}$	66.02	**.78**$_{(.78)}$	**.55**$_{(.55)}$	**.45**$_{(.45)}$
DKP-AOM	11	19	2.53	**.39**$_{(.92)}$	.03$_{(.08)}$	.01$_{(.04)}$	4.23	.50$_{(.99)}$	.01$_{(.02)}$	.01$_{(.01)}$
GMap	2069	21	1.69	.37$_{(.80)}$	.03$_{(.06)}$	.01$_{(.03)}$	3.13	.67$_{(.98)}$	.01$_{(.02)}$	.01$_{(.01)}$
LogMap-C	56	19	1.41	.38$_{(.90)}$	.03$_{(.09)}$	.02$_{(.04)}$	3.68	.35$_{(.56)}$	.01$_{(.03)}$	.01$_{(.01)}$
LogMapLt	13	19	1.29	**.39**$_{(.91)}$	.04$_{(.08)}$	.02$_{(.04)}$	3.70	.32$_{(.57)}$	.01$_{(.03)}$	.01$_{(.01)}$
Mamba*	297	21	1.52	.36$_{(.78)}$	.06$_{(.13)}$	.03$_{(.07)}$	3.68	.48$_{(.99)}$	.02$_{(.05)}$	.01$_{(.03)}$
RSDLWB	14	45	30.71	.01$_{(.01)}$	.01$_{(.01)}$	.01$_{(.01)}$	43.71	.20$_{(.20)}$	.11$_{(.11)}$	.08$_{(.08)}$

Table 14. MultiFarm aggregated results per matcher, for each type of matching task – different ontologies (i) and same ontologies (ii). Time is measured in minutes (for completing the 45×25 matching tasks). Tools marked with an * have been executed in a different setting. #pairs indicates the number of pairs of languages for which the tool is able to generated (non empty) alignments. Size indicates the average of the number of generated correspondences for the tests where an (non empty) alignment has been generated. Two kinds of results are reported: those do not distinguishing empty and erroneous (or not generated) alignments and those – indicated between parenthesis – considering only non empty generated alignments for a pair of languages.

non-specific systems are not able to deal with all pairs of languages, in particular those involving Arabic, Chinese and Russian. Instead, they take advantage of the similarities in the vocabulary of some languages, in the absence of specific strategies. This can be corroborated by the fact that most of them generate their best F-measure for the pairs es-pt (followed by de-en): CroMatcher (es-pt .28, de-en .23), DKP-AOM (es-pt .25, de-en .22), GMap (es-pt .21, fr-nl .20), LogMap-C (es-pt .26, de-en .18), LogMapLt (es-pt .25, de-en .22), and Mamba (es-pt .29, en-nl .23, de-en .22). This behavior has been also observed last year. On the other hand, although it is likely harder to find correspondences between cz-pt than es-pt, for some non-specific systems this pair is present in their top-3 F-measure (with the exception of Mamba).

For the group of systems implementing cross-lingual strategies, some pairs involving Czech (cz-en, cz-es, cz-pt, cz-de, cz-ru) are again present in the top-5 F-measure of 4 systems (out of 5, the exception is LYAM++): AML – cz-en (.63), cz-ru (.62), cz-es (.61), cz-nl (.60), en-es (.59), CLONA – es-ru (.53), cz-es (.51), es-pt (.51), cz-en (.50) and cz-ru (.49), LogMap – cz-de (.55), cz-pt (.54), cz-ru (.53), cz-nl and cz-en (.52), XMap – cz-es (.52), cz-pt (.50), en-es (.48), cz-ru (.45), and de-es (.45). LYAM++ is the exception, once it was not able to generate alignments for some of these pairs : es-fr (.56), en-es (.53), es-pt (.52), en-ru (.52) and en-fr (.52). A different behavior is observed for the tasks of type (ii), for which these systems perform better for the pairs en-pt, es-fr, en-fr, de-en and es-pt. The exception is LogMap (es-ru, es-nl and fr-nl).

Edas and Ekaw based evaluation. Table 15 presents the aggregated results for the matching tasks involving edas and ekaw ontologies. LYAM++ has participated only in the open test. The overall results here are close to what has been observed for the open evaluation. For both types of tasks, LogMap outperforms all systems in terms of precision and AML in terms of F-measure. Both of them required more time for finishing the tasks due to the fact that new translations were computed on the fly (for Italian).

Looking at the overall results of non-specific systems, for the cases of type (i), DKP-OAM still generates good precision values but has been outperformed by GMap and Mamba. For the cases of type (ii), CroMatcher corroborates the good results obtained by its structural strategy, while LogMap-C and LogMap-Lite decrease their precision, considerably increasing the number of generated correspondences (in particular for the edas-edas task).

With respect to the pairs of languages for the test cases of type (i), although the overall results remain relatively stable, new pairs of languages take place in the top-3 F-measure. For non specific systems, it is the case for the pairs es-it and it-pt : CroMatcher (es-it .25, it-pt .25, en-it .24, and en-nl .21), DKP-AOM (es-pt .20, de-en .20, it-pt .17, es-it .16), GMap (it-pt .31, en-it .25, en-fr .19), LogMap-C (de-en .23, es-pt .21, it-pt .20, es-it .19), LogMapLt (de-en .20, es-pt .20, it-pt .17, es-it .16), and Mamba (de-en .27, en-it .26, en-nl .25, it-pt .24). For the group of systems implementing cross-lingual strategies, this fact has been observed for 2 (AML and XMAP) out of 4 systems. For those systems, some pairs involving Czech (cn-cz, cz-de, cz-en ou cz-ru) are again present in the top-5 F-measure of 3 out of 4 systems: AML (es-it .58, en-pt .58 en-nl .57 cz-en .57 nl-pt .57 es-nl .56, cz-nl .55, en-es .55, cz-es .54), CLONA (cn-cz .38, cz-pt .38, de-pt .38, de-en .37, fr-pt .37, pt-ru .36, es-pt .36, es-ru .35, fr-ru .35, cz-de .35), LogMap (en-nl .53, en-pt .51, cz-en .49, en-ru .48, cz-nl .46, cz-ru .46). The exception is XMAP (nl-pt .53, nl-ru .43, it-pt .41, pt-ru .37, fr-ru .37). Finally, with respect to type (ii), the pair it-pt appears in the top-3 F-measure of AML and CLONA.

Comparison with previous campaigns. In the first year of evaluation of MultiFarm (2011.5 campaign), 3 participants (out of 19) implemented specific techniques. In 2012, we counted on 7 systems (out of 24). We had the same number of participants in 2013. In 2014, this number decreased considerably (3 systems). All of them participate this year (AML, LogMap and XMap) and we count on two new participants (LYAM++, in fact an extension to YAM++ that has participated in previous campaigns, and CLONA). Comparing the previous F-measure results (on the same basis, i.e., open data set and tasks of type (ii) and excluding Arabic translations[14]), this year AML (.54) remains stable with respect to 2014 and outperforms the best system in 2013 and 2012 – YAM++ (.40) – while LogMap (.42) slightly improves the results obtained in 2014 (.40). While LogMapLt and LogMap-C improved precision (.15 up to .39), RSDLWB decreased in recall. In overall, the performance of the systems remain stable over these last two years.

[14] The French translations have been revised. This revision does not seem to have a major impact on the overall results. However, this impact has not been deeply measured, what has to be done with respect to tool versions used in the OAEI 2014.

			Type (i) – 22 tests per pair				Type (ii) – 2 tests per pair			
System	Time	#pairs	Size	Prec.	F-m.	Rec.	Size	Prec.	F-m.	Rec.
AML	128	55	13.33	.52$_{(.52)}$	**.47**$_{(.47)}$	**.42**$_{(.42)}$	68.62	.93$_{(.93)}$	**.64**$_{(.64)}$	**.49**$_{(.49)}$
CLONA*	931	55	9.62	.40$_{(.40)}$	.29$_{(.29)}$	.23$_{(.23)}$	61.98	.88$_{(.88)}$	.57$_{(.57)}$	.42$_{(.42)}$
LogMap*	253	55	7.43	**.71**$_{(.71)}$	.38$_{(.38)}$	.27$_{(.27)}$	52.69	**.97**$_{(.97)}$	.44$_{(.44)}$	.30$_{(.30)}$
LYAM++**	-	-	-	-	-	-	-	-	-	-
XMap	11877	52	182.55	.14$_{(.15)}$	.13$_{(.13)}$	.17$_{(.18)}$	285.53	.40$_{(.44)}$	.22$_{(.24)}$	.19$_{(.21)}$
CroMatcher	297	55	13.53	.32$_{(.32)}$	.09$_{(.09)}$	.06$_{(.06)}$	75.08	**.81**$_{(.81)}$	**.54**$_{(.54)}$	**.44**$_{(.44)}$
DKP-AOM	20	24	2.58	.43$_{(.98)}$	.04$_{(.09)}$	.02$_{(.05)}$	4.37	.49$_{(1.0)}$	.02$_{(.03)}$	.01$_{(.01)}$
GMap	2968	27	1.81	.45$_{(.92)}$	.05$_{(.11)}$	.03$_{(.06)}$	4.4	.49$_{(.99)}$	.02$_{(.05)}$	.01$_{(.02)}$
LogMap-C	73	26	1.24	.38$_{(.81)}$	.05$_{(.10)}$	.03$_{(.05)}$	93.69	.02$_{(.04)}$	.01$_{(.03)}$	.01$_{(.02)}$
LogMapLt	17	25	1.16	.36$_{(.78)}$	.04$_{(.09)}$	.02$_{(.05)}$	94.5	.02$_{(.04)}$	.01$_{(.03)}$	.01$_{(.02)}$
Mamba*	383	28	1.81	**.48**$_{(.93)}$	.08$_{(.15)}$	.04$_{(.09)}$	3.74	.59$_{(.99)}$	.03$_{(.05)}$	.01$_{(.02)}$
RSDLWB	19	55	32.12	.01$_{(.01)}$	.01$_{(.01)}$	.01$_{(.01)}$	43.31	.19$_{(.10)}$	.10$_{(.10)}$	.06$_{(.06)}$

Table 15. MultiFarm aggregated results per matcher for the edas and ekaw based evaluation, for each type of matching task – different ontologies (i) and same ontologies (ii). Time is measured in minutes (for completing the 55×24 matching tasks).

7.4 Conclusion

As expected, systems implementing specific methods for dealing with ontologies in different languages outperform non specific systems. Overall, the results remain stable with respect to the last campaigns (F-measure around .54), with precision being privileged with respect to recall. While some systems can take advantage of the ontology structure to overcome the lack of cross-lingual strategies, some of them are not able to deal at all with certain group of languages (Arabic, Chinese, Russian). Still, cross-lingual approaches are mainly based on translation strategies and the combination of other resources (like cross-lingual links in Wikipedia, BabelNet, etc.) and strategies (machine learning, indirect alignment composition) remains underexploited.

8 Interactive matching

The interactive matching track was organized at OAEI 2015 for the third time. The goal of this evaluation is to simulate interactive matching [29], where a human expert is involved to validate correspondences found by the matching system. In the evaluation, we look at how interacting with the user improves the matching results. Currently, this track does not evaluate the user experience or the user interfaces of the systems.

8.1 Experimental setting

The SEALS client was modified to allow interactive matchers to ask an oracle. The interactive matcher can present a correspondence to the oracle, which then tells the system whether the correspondence is right or wrong. A request is considered distinct if one of the concepts or the relationship in a correspondence have changed in comparison

with previous requests. This year, in addition to emulating the perfect user, we also consider domain experts with variable error rates which reflects a more realistic scenario where a user does not necessarily provide a correct answer. We experiment with three different error rates: 0.1, 0.2 and 0.3. The errors were randomly introduced into the reference alignment with given rates.

The evaluations of the conference and anatomy datasets were run on a server with 3.46 GHz (6 cores) and 8GB RAM allocated to the matching system. Each system was run three times and the final result of a system for each error rate represents the average of these runs. This is the same configuration which was used in the non-interactive version of the anatomy track and runtimes in the interactive version of this track are therefore comparable. For the conference dataset with the ra1 alignment, we considered macro-average of precision and recall of different ontology pairs, while the number of interactions represent the total number of interactions in all tasks. Finally, the three runs are averaged. The largebio dataset evaluation (each system was run one time) was run on a Ubuntu Laptop with an Intel Core i7-4600U CPU @ 2.10GHz x 4 and allocating 15GB of RAM.

8.2 Data sets

In this third edition of the Interactive track we use three OAEI datasets, namely conference, anatomy and Large Biomedical Ontologies (largebio) dataset. From the conference dataset we only use the test cases for which an alignment is publicly available (altogether 21 alignments/tasks). The anatomy dataset includes two ontologies (1 task), the Adult Mouse Anatomy (AMA) ontology and a part of the National Cancer Institute Thesaurus (NCI) describing the human anatomy. Finally, largebio consists of 6 tasks with different sizes ranging from tens to hundreds of thousands classes and aims at finding alignments between the Foundational Model of Anatomy (FMA), SNOMED CT, and the National Cancer Institute Thesaurus (NCI).

8.3 Systems

Overall, four systems participated in the Interactive matching track: AML, JarvisOM, LogMap, and ServOMBI. The systems AML and LogMap have been further developed compared to last year, the other two participated in this track for the first time. All systems participating in the Interactive track support both interactive and non-interactive matching. This allows us to analyze how much benefit the interaction brings for the individual system.

The different systems involve the user in different points of the execution and use the user input in different ways. Therefore, we describe how the interaction is done by each system. AML starts interacting with the user during the selection and repairing phases (for the largebio task only non-interactive repair is employed) at the end of the matching process. The user input is employed to filter correspondences included in the final alignment and AML does not generate new correspondences nor adjust matching parameters based on it. AML avoids asking the same question more than once by keeping track of already asked questions and uses a query limit and other strategies to stop asking the user and reverts to non-interactive mode.

JarvisOM is based on an active learning strategy known as query-by-committee. In this strategy, informative instances are those where the committee members (classifiers; 3 in this campaign) disagree most. Sample entity pairs are selected using the heuristic of the Farthest First algorithm in order to initialize the classifiers committee. At every iteration JarvisOM asks the user for pairs of entities that have the highest value for the vote entropy measure (disagreement between committee members) and lower average euclidean distance. In the last iteration, the classifiers committee is used to generate the alignment between the ontologies.

ServOMBI uses various similarity measures during the Terminological phase after which the results are presented to the user. The user input is then used in the Contextual phase which employs machine learning techniques. The user is then asked again to validate the newly generated candidate correspondences (according to given threshold). At the end, an algorithm is run to determine the correspondences in the final alignment.

LogMap generates candidate correspondences first and then employs different techniques (lexical, structural and reasoning-based) to discard some of them during the Assessment phase. During this phase in the interactive mode it interacts with the user and presents to him/her those correspondences which are not clear-cut cases.

8.4 Results for the Anatomy dataset

Tables 16, 17, 18 and 19 present the results for the Anatomy dataset with four different error rates. The first three columns in each of the tables present the adjusted results obtained in this track (in the adjusted results the trivial correspondences in the oboInOwl- namespace have been removed as well as correspondences expressing relations different from equivalence). We adjust the results in order to enable the comparison between the measures obtained in this and the non-interactive Anatomy track. The measure recall+ indicates the amount of detected non-trivial correspondences (trivial correspondences are those with the same normalized label). The precision, recall and F-measure columns at the right end of the tables present the results as calculated by the SEALS client prior to the adjustment. The last three columns contain the evaluation results "according to the oracle", meaning against the oracle's alignment, i.e., the reference alignment as modified by the randomly introduced errors. Figure 3 shows the time intervals between the questions to the user/oracle for the different systems and error rates for the three runs (the runs are depicted with different colors).

We first compare the performance of the four systems with an all-knowing oracle (0.0 error rate - Table 16), in terms of precision, recall and F-measure, to the results obtained in the non-interactive Anatomy track (these are the first 6 columns in the corresponding tables). The effect of introducing interactions with the oracle/user is mostly pronounced for the precision measure (except for JarvisOM). In the Interactive track (and 0.0 error rate) the precision for all four systems improves and, consequently, so does the F-measure. At the same time the recall improves for AML and JarvisOM and does not change for LogMap and ServOMBI. AML achieves the best F-measure and recall among the four with a perfect oracle. Out of all systems, JarvisOM displays the largest improvements when user interactions are brought in—the F-measure improves almost 4,5 times together with the recall which improves 6 times and the precision goes

Tool	Prec.	F-m.	Rec.	Rec.+	Size	Tot. Reqs.	Dist. Reqs.	TP	TN	FP	FN	Time	Prec.	F-m.	Rec.	Prec. Oracle	F-m. Oracle	Rec. Oracle
AML	0.97	0.96	0.95	0.88	1491.0	312.0	312.0	73.0	239.0	0.0	0.0	49	0.97	0.96	0.95	0.97	0.96	0.95
JarvisOM	0.87	0.76	0.67	0.15	1168.0	7.0	7.0	4.0	3.0	0.0	0.0	213	0.86	0.75	0.67	0.86	0.75	0.67
LogMap	0.99	0.91	0.85	0.60	1298.0	590.0	590.0	287.0	303.0	0.0	0.0	24	0.98	0.91	0.85	0.98	0.91	0.85
ServOMBI	1.00	0.76	0.62	0.10	935.0	2136.0	1128.0	955.0	173.0	0.0	0.0	711	1.00	0.76	0.62	1.00	0.76	0.62

Table 16. Anatomy dataset – perfect oracle

Tool	Prec.	F-m.	Rec.	Rec.+	Size	Tot. Reqs.	Dist. Reqs.	TP	TN	FP	FN	Time	Prec.	F-m.	Rec.	Prec. Oracle	F-m. Oracle	Rec. Oracle
AML	0.96	0.95	0.95	0.86	1502.0	317.3	317.3	66.3	218.0	23.0	10.0	45	0.96	0.95	0.95	0.97	0.96	0.95
JarvisOM	0.76	0.68	0.67	0.22	1467.7	7.0	7.0	3.3	3.0	0.3	0.3	214	0.76	0.68	0.67	0.76	0.68	0.67
LogMap	0.97	0.89	0.83	0.57	1306.0	609.0	609.0	261.3	288.3	33.7	25.7	25	0.96	0.89	0.83	0.96	0.89	0.83
ServOMBI	1.00	0.71	0.55	0.08	842.7	2198.7	1128.0	857.3	156.3	16.7	97.7	563	1.00	0.71	0.55	1.00	0.74	0.59

Table 17. Anatomy dataset – error rate 0.1

Tool	Prec.	F-m.	Rec.	Rec.+	Size	Tot. Reqs.	Dist. Reqs.	TP	TN	FP	FN	Time	Prec.	F-m.	Rec.	Prec. Oracle	F-m. Oracle	Rec. Oracle
AML	0.94	0.94	0.94	0.85	1525.0	321.7	321.7	66.3	186.7	52.3	16.3	47	0.94	0.94	0.94	0.97	0.96	0.95
JarvisOM	0.53	0.60	0.71	0.38	2045.3	8.0	8.0	4.7	1.0	1.3	1.0	214	0.53	0.60	0.71	0.53	0.60	0.71
LogMap	0.95	0.88	0.82	0.56	1311.7	630.0	630.0	233.0	274.0	69.0	54.0	24	0.95	0.88	0.82	0.95	0.88	0.81
ServOMBI	0.99	0.66	0.49	0.08	757.0	2257.0	1128.0	767.3	131.3	41.7	187.7	571	0.99	0.66	0.49	1.00	0.71	0.55

Table 18. Anatomy dataset – error rate 0.2

Tool	Prec.	F-m.	Rec.	Rec.+	Size	Tot. Reqs.	Dist. Reqs.	TP	TN	FP	FN	Time	Prec.	F-m.	Rec.	Prec. Oracle	F-m. Oracle	Rec. Oracle
AML	0.93	0.93	0.94	0.84	1526.0	306.0	306.0	54.0	168.7	61.3	22.0	48	0.93	0.93	0.94	0.97	0.96	0.95
JarvisOM	0.51	0.49	0.53	0.25	1501.7	7.3	7.3	4.0	1.7	1.0	0.7	214	0.51	0.49	0.53	0.51	0.49	0.53
LogMap	0.94	0.88	0.82	0.54	1317.0	663.0	663.0	200.7	270.7	105.3	86.3	24	0.94	0.87	0.82	0.92	0.86	0.80
ServOMBI	0.99	0.60	0.43	0.07	658.3	2329.7	1128.3	663.3	129.0	44.3	291.7	447	0.99	0.60	0.43	1.00	0.68	0.52

Table 19. Anatomy dataset – error rate 0.3

up 2,5 times. The size of the alignment generated by the system also grows around 2,5 times.

With the introduction of an erroneous oracle/user and moving towards higher error rates, system performance, obviously, starts to slightly deteriorate in comparison to the all-knowing oracle. However, the changes in the error rates influence the four systems differently in comparison to the non-interactive results. While the AML performance with an all-knowing oracle is better on all measures with respect to the non-interactive results, the F-measure drops in the 0.2 and 0.3 cases (Tables 18 and 19), while the recall stays higher than the non-interactive results for all error rates. LogMap behaves similarly—the F-measure in the 0.2 and 0.3 cases drops below the non-interactive results, while the precision stays higher in all error rates. ServOMBI performance in terms of F-measure and Recall drops below the non-interactive results already in the 0.1 case (Table 17), but the precision is higher in all cases. In contrast JarvisOM still performs better in the 0.3 case on all measures than in the non-interactive Anatomy track where it achieved very low values for all measures. It is also worth noting the large drop in precision (around 35 percentage points) for JarvisOM with the growing error rates in comparison to the other three systems where the drop in precision is between 1 to 5 percentage points. This could be explained by the fact that JarvisOM asks only few questions and is therefore very sensitive to false positives and false negatives. Another interesting observation is that, with the exception of AML, the performance of the systems also declines as the error increases with regard to the oracle's reference (i.e., the reference as modified by the errors introduced in the oracle). This means that the impact of the errors is linear for AML (i.e., one erroneous response from the oracle, leads to only one error from AML) but supralinear for the other systems.

AML also shows stable performance in connection to the size of the alignment and the number of (distinct) requests to the oracle generated with different error rates. As discussed it does not present the same question again to the user. The same observation regarding the unique requests applies to JarvisOM and LogMap as well. JarvisOM uses very few requests to the oracle and this number is stable across the different error rates. Another notable difference is the varying size of the alignment generated by JarvisOM which almost doubles in the 0.2 case comparing to the all-knowing oracle. The number of requests grows with the error rate for LogMap together with a slight grow in the alignment size. As we noted above ServOMBI asks the user for every correspondence found and the number of distinct requests for ServOMBI stays stable for the different rates. The total number of requests is almost double the distinct ones but at the same time the size of the alignment drops when introducing higher error rates. The run times between the different error rates slightly change for AML while there is no significant change for LogMap and JarvisOM. The ServOMBI run time decreases with the increase of the error rate. In comparison to the non-interactive track, LogMap's and JarvisOM's run times do not change and AML's run time changes between 10 to 20 %. ServOMBI run time is higher in the non-interactive track.

For an interactive system the time intervals at which the user is involved in an interaction are important. Figure 3 presents a comparison between the systems regarding the time periods at which the system presents a question to the user. Across the three runs and different error rates the AML and LogMap request intervals are around 1 and 0

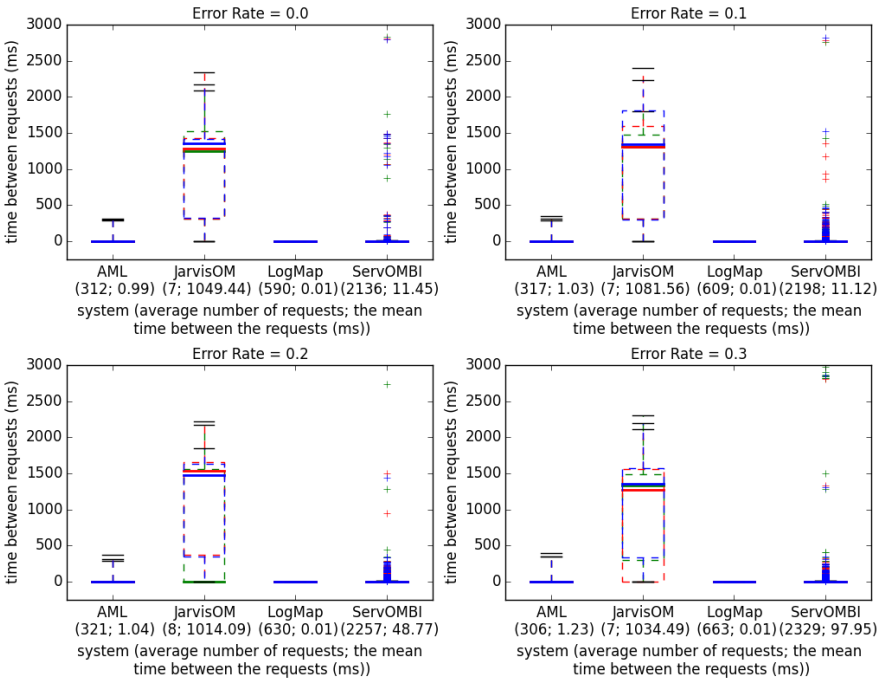

Fig. 3. The Y axis depicts the time intervals between the requests to the user/oracle (whiskers: Q1-1,5IQR, Q3+1,5IQR, IQR=Q3-Q1). The labels under the system names show the average number of requests and the mean time between the requests for the three runs.

milliseconds respectively. On the other hand, while the requests periods for ServOMBI are under 10 ms in most of the cases we see that there are some outliers requiring more than a second. Furthermore a manual inspection of the intervals showed that in several cases it takes more than 10 seconds between the questions to the user and in one extreme case—250 seconds. It can also be seen that the requests intervals for this system increase at the last 50–100 questions. JarvisOM displays a delay in its requests in comparison to the other systems. The average interval at which a question is presented to the user is 1 second with about half of the requests to the user taking more than 1,5 seconds. However it issues the quetions during the alignemnt process and not as a post processing step.

The take away of this analyses is the large improvement for JarvisOM in all measures and error rates with respect to its non-interactive results. The growth of the error rate impacts different measures in the different systems. The effect of introducing interactions with the oracle/user is mostly pronounced for the precision measure - the precision for all systems (except AML) in the different error rates is higher than their precision in the evaluation of the non-interactive Anatomy track.

8.5 Results for the conference dataset

Tables 20, 21, 22 and 23 below present the results for the Conference dataset with four different error rates. The "Precision Oracle", "Recall Oracle" and "F-measure Oracle" columns contain the evaluation results "according to the oracle", meaning against the oracle's alignment (i.e., the reference alignment as modified by the randomly introduced errors). Figure 4 shows the average requests intervals per task (21 tasks in total per run) between the questions to the user/oracle for the different systems and error rates for all tasks and the three runs (the runs are depicted with different colors). The first number under the system names is the average number of requests and the second number is the average period of the average requests intervals for all tasks and runs.

We first focus on the performance of the systems with an all-knowing oracle (Table 20). In this case, all systems improve their results compared to the non-interactive version of the Conference track. The biggest improvement in F-measure is achieved by ServOMBI with 23 percentage points. Other systems also show substantial improvements, AML improves the F-measure by 8, JarvisOM by 13 and LogMap by around 4 percentage points. Closer inspection shows that for different systems the improvement of F-measure can be attributed to different factors. For example, in the case of ServOMBI and LogMap interaction with the user improved precision while recall experienced only slight improvement. On the other hand, JarvisOM improved recall substantially while keeping similar level of precision. Finally, AML improved precision by 10 and recall by 6 percentage points which contributed to a higher F-measure.

As expected, the results start deteriorating when introducing the error in the oracle's answers. Interestingly, even with the error rate of 0.3 (Table 23) most systems perform similar (with respect to the F-measure) to their non-interactive version. For example, AML's F-measure in the case with 0.3 error rate is only 1 percentage point worse than the non-interactive one. The most substantial difference is in the case of ServOMBI with an oracle with the error rate of 0.3 where the system achieves around 5 percentage points worse result w.r.t. F-measure than in the non-interactive version. Again closer inspection shows that different systems are affected in different ways when errors are introduced. For example, if we compare the 0.0 and 0.3 case, we can see that for AML, precision is affected by 11 and recall by 6 percentage points. In the case of JarvisOM, precision drops by 19 while recall drops by only 4 percentage points. LogMap is affected in a similar manner and its precision drops by 9 while the recall drops by only 3 percentage points. Finally, the most substantial change is in the case of ServOMBI where the precision drops from 100% to 66% and the recall shows a drop of 22 percentage points. Like in the Anatomy dataset, LogMap and ServOMBI also show a drop in performance in relation to the oracle's reference with the increase of the error rate, which indicates a supralinear impact of the errors. AML again shows a constant performance that reflects a linear impact of the errors. Surprisingly, JarvisOM also shows a constant performance, which is a different behavior than in the anatomy case.

When it comes to the number of request to the oracle, 3 out of 4 systems do around 150 requests while ServOMBI does most requests, namely 550. AML, JarvisOM and LogMap do not repeat their requests while around 40% of requests done by ServOMBI are repeated requests. Across the three runs and different error rates the AML and LogMap mean times between requests for all tasks are less than 3 ms. On the other

Tool	Prec.	Prec. non	F-m.	F-m. non	Rec.	Rec. non	Prec. Oracle	F-m. Oracle	Rec. Oracle	Tot. Reqs.	Dist. Reqs.	TP	TN	FP	FN	Time
AML	0.94	0.84	0.82	0.74	0.72	0.66	0.94	0.82	0.72	147.0	147.0	53.0	94.0	0.0	0.0	28
JarvisOM	0.81	0.84	0.65	0.52	0.55	0.37	0.81	0.65	0.55	154.0	154.0	38.0	116.0	0.0	0.0	39
LogMap	0.87	0.80	0.72	0.68	0.62	0.59	0.87	0.72	0.62	157.0	157.0	52.0	105.0	0.0	0.0	27
ServOMBI	1.00	0.56	0.79	0.57	0.65	0.59	1.00	0.79	0.65	535.0	295.0	156.0	139.0	0.0	0.0	50

Table 20. Conference dataset – perfect oracle

Tool	Prec.	Prec. non	F-m.	F-m. non	Rec.	Rec. non	Prec. Oracle	F-m. Oracle	Rec. Oracle	Tot. Reqs.	Dist. Reqs.	TP	TN	FP	FN	Time
AML	0.91	0.84	0.79	0.74	0.71	0.66	0.94	0.82	0.73	147.3	147.3	48.3	85.3	8.7	5.0	27
JarvisOM	0.73	0.84	0.61	0.52	0.53	0.37	0.77	0.64	0.55	154.0	154.0	34.3	107.0	10.3	2.3	38
LogMap	0.83	0.80	0.69	0.68	0.60	0.59	0.84	0.69	0.59	157.7	157.7	45.7	93.3	12.3	6.3	27
ServOMBI	0.89	0.56	0.70	0.57	0.57	0.59	1.00	0.78	0.64	555.3	299.3	137.7	126.3	16.7	18.7	51

Table 21. Conference dataset – error rate 0.1

Tool	Prec.	Prec. non	F-m.	F-m. non	Rec.	Rec. non	Prec. Oracle	F-m. Oracle	Rec. Oracle	Tot. Reqs.	Dist. Reqs.	TP	TN	FP	FN	Time
AML	0.87	0.84	0.77	0.74	0.69	0.66	0.94	0.82	0.73	149.0	149.0	45.0	76.3	17.3	10.3	27
JarvisOM	0.67	0.84	0.58	0.52	0.52	0.37	0.77	0.65	0.56	155.0	155.0	28.7	97.3	22.7	6.3	38
LogMap	0.81	0.80	0.69	0.68	0.59	0.59	0.81	0.68	0.58	158.7	158.7	40.0	84.7	22.0	12.0	27
ServOMBI	0.80	0.56	0.61	0.57	0.50	0.59	1.00	0.77	0.62	554.7	295.7	122.0	110.7	29.0	34.0	50

Table 22. Conference dataset – error rate 0.2

Tool	Prec.	Prec. non	F-m.	F-m. non	Rec.	Rec. non	Prec. Oracle	F-m. Oracle	Rec. Oracle	Tot. Reqs.	Dist. Reqs.	TP	TN	FP	FN	Time
AML	0.83	0.84	0.73	0.74	0.66	0.66	0.94	0.82	0.73	148.7	148.7	35.3	68.0	24.7	20.7	27
JarvisOM	0.62	0.84	0.56	0.52	0.51	0.37	0.74	0.65	0.58	154.3	154.3	24.3	88.0	32.0	10.0	39
LogMap	0.78	0.80	0.67	0.68	0.59	0.59	0.79	0.66	0.57	154.0	154.0	37.7	70.7	31.3	14.3	27
ServOMBI	0.66	0.56	0.52	0.57	0.43	0.59	1.00	0.77	0.63	589.7	308.0	105.0	103.7	48.0	51.3	50

Table 23. Conference dataset – error rate 0.3

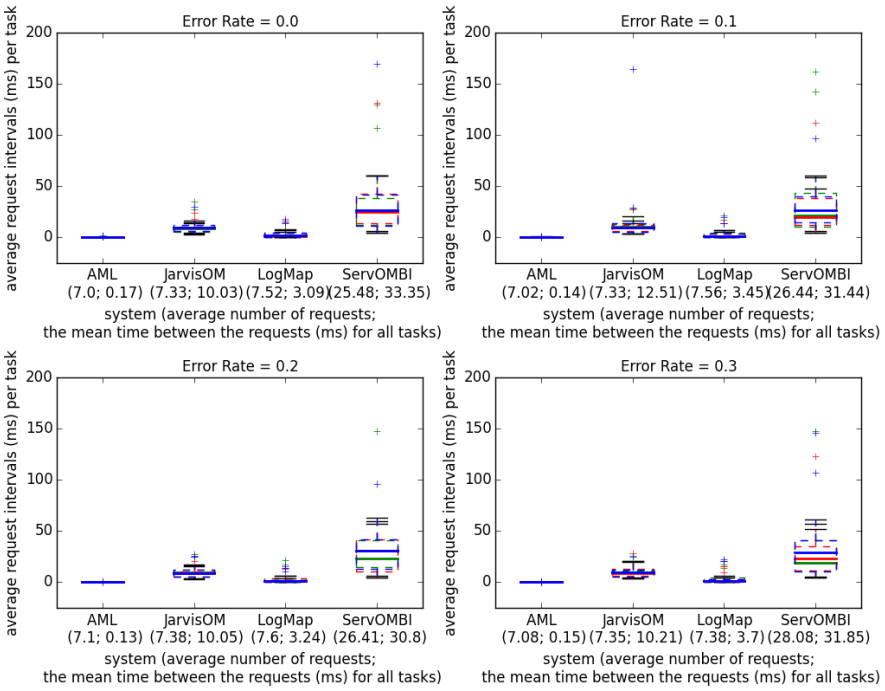

Fig. 4. The Y axis depicts the average time between the requests per task in the Conference dataset (whiskers: Q1-1,5IQR, Q3+1,5IQR, IQR=Q3-Q1). The labels under the system names show the average number of requests and the mean time between the requests (calculated by taking the average of the average request intervals per task) for the three runs and all tasks.

hand, mean time between requests for ServOMBI and JarvisOM are around 30 and 10 ms respectively. While in most cases there is little to no delay between requests, there are some outliers. These are most prominent for ServOMBI where some requests were delayed for around 2 seconds which is substaintally longer than the mean.

This year we have two systems, AML and LogMap, which competed in the last year's campaign. When comparing to the results of last year (perfect oracle), AML improved its F-measure by around 2 percentage points. This increase can be accounted to increased precision (increase of around 3 percentage points). On the other hand, LogMap shows a slight decrease in recall and precision, and hence, in F-measure.

8.6 Results for the largebio dataset

Tables 24, 25, 26 and 27 below present the results for the largebio dataset with four different error rates. The "precision oracle", "recall oracle" and "F-measure oracle" columns contain the evaluation results "according to the oracle", meaning against the oracle's alignment, i.e., the reference alignment as modified by the randomly introduced errors. Figure 5 shows the average requests intervals per task (6 tasks in total) between

the questions to the user/oracle for the different systems and error rates for all tasks and a single runs. The first number under the system names is the average number of requests and the second number is the average period of the average requests intervals for all tasks in the run.

Of the four systems participating in this track this year, only AML and LogMap were able to complete the full largebio dataset. ServOMBI was only able to match the FMA-NCI small fragments and FMA-SNOMED small fragments, whereas JarvisOM was unable to complete any of the tasks. Therefore, ServOMBI's results are partial, and not directly comparable with those of the other systems (marked with * in the results table and Figure 5).

With an all-knowing oracle (Table 24), AML, LogMap and ServOMBI all improved their performance in comparison with the non-interactive version of the largebio track. The biggest improvement in F-measure was achieved by LogMap with 4, followed by AML with 3, then ServOMBI with 2 percentage points. AML showed the greatest improvement in terms of recall, but also increased its precision substantially; LogMap had the greatest improvement in terms of precision, but also showed a significant increase in recall; and ServOMBI improved essentially only with regard to precision, obtaining 100% as in the other datasets.

The introduction of (simulated) user errors had a very different effect on the three systems: AML shows a slight drop in performance of 3 percentage points in F-measure between 0 and 0.3 error rate (Table 27), and is only slightly worse than its non-interactive version at 0.3 error rate; LogMap shows a more pronounced drop of 6 percentage points in F-measure; and ServOMBI shows a substantial drop of 17 percentage points in F-measure. Unlike in the other datasets, all systems are affected significantly by the error with regard to both precision and recall. Like in the other datasets, AML shows a constant performance in relation to the oracle's reference, indicating a linear impact of the errors, whereas the other two systems decrease in performance as the error increases, indicating a supralinear impact of the errors.

Regarding the number of request to the oracle, AML was the more sparing system, with only 10,217, whereas LogMap made almost three times as many requests (27,436). ServOMBI was again the more inquisitive system, with 21,416 requests on only the two smallest tasks in the dataset (for comparison, AML made only 1,823 requests on these two tasks and LogMap made 6,602). As in the other datasets, ServOMBI was the only system to make redundant requests to the oracle. Interestingly, both LogMap and ServOMBI increased the number of requests with the error, whereas AML had a constant number of requests. Figure 5 presents a comparison between the systems regarding the average time periods for all tasks at which the system presents a question to the user. Across the different error rates the average requests intervals for all tasks for AML and LogMap are around 0 millisecond. For ServOMBI they are slightly higher (25 milliseconds on average) but a manual inspection of the results shows some intervals larger than 1 second (often those are between some of the last requests the system performs).

8.7 Discussion

This year is the first time we have considered a non-perfect domain expert, i.e., a domain expert which can provide wrong answers. As expected, the performance of the

Tool	Prec.	Prec. non	F-m.	F-m. non	Rec.	Rec. non	Prec. Oracle	F-m. Oracle	Rec. Oracle	Tot. Reqs.	Dist. Reqs.	TP	TN	FP	FN	Time
AML	0.94	0.91	0.85	0.82	0.77	0.75	0.94	0.85	0.77	10217	10217	5126	5091	0	0	2877
LogMap	0.97	0.90	0.83	0.79	0.73	0.71	0.97	0.83	0.73	27436	27436	17050	10386	0	0	3803
ServOMBI*	1.00	0.97	0.85	0.83	0.74	0.74	1.00	0.85	0.74	21416	9424	8685	739	0	0	726

Table 24. Largebio dataset – perfect oracle

Tool	Prec.	Prec. non	F-m.	F-m. non	Rec.	Rec. non	Prec. Oracle	F-m. Oracle	Rec. Oracle	Tot. Reqs.	Dist. Reqs.	TP	TN	FP	FN	Time
AML	0.93	0.91	0.84	0.82	0.76	0.75	0.94	0.85	0.77	10217	10217	4624	4658	485	450	2913
LogMap	0.94	0.90	0.80	0.79	0.70	0.71	0.94	0.80	0.70	28890	28890	15753	10659	1181	1297	3963
ServOMBI*	1.00	0.97	0.80	0.83	0.67	0.74	1.00	0.83	0.72	22920	9502	8063	726	85	628	695

Table 25. Largebio dataset – error rate 0.1

Tool	Prec.	Prec. non	F-m.	F-m. non	Rec.	Rec. non	Prec. Oracle	F-m. Oracle	Rec. Oracle	Tot. Reqs.	Dist. Reqs.	TP	TN	FP	FN	Time
AML	0.92	0.91	0.82	0.82	0.75	0.75	0.94	0.85	0.77	10217	10217	4196	4081	1049	891	2930
LogMap	0.92	0.90	0.78	0.79	0.68	0.71	0.91	0.77	0.68	30426	30426	14286	10707	2669	2764	3912
ServOMBI*	0.99	0.97	0.74	0.83	0.59	0.74	1.00	0.81	0.69	23968	9541	7431	661	192	1257	713

Table 26. Largebio dataset – error rate 0.2

Tool	Prec.	Prec. non	F-m.	F-m. non	Rec.	Rec. non	Prec. Oracle	F-m. Oracle	Rec. Oracle	Tot. Reqs.	Dist. Reqs.	TP	TN	FP	FN	Time
AML	0.91	0.91	0.82	0.82	0.75	0.75	0.94	0.85	0.77	10217	10217	3737	3637	1537	1306	2959
LogMap	0.90	0.90	0.77	0.79	0.68	0.71	0.87	0.74	0.65	31504	31504	13035	10147	4307	4015	3874
ServOMBI*	0.98	0.97	0.68	0.83	0.52	0.74	1.00	0.79	0.66	25580	9600	6818	652	256	1874	618

Table 27. Largebio dataset – error rate 0.3

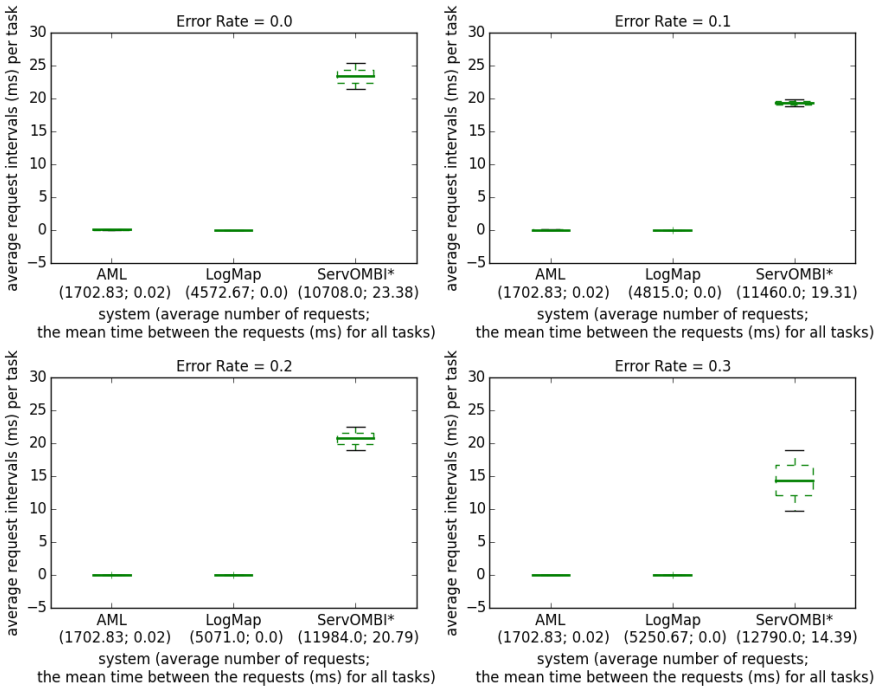

Fig. 5. The Y axis depicts the average time between the requests per task in the largebio dataset (6 tasks) (whiskers: Q1-1,5IQR, Q3+1,5IQR, IQR=Q3-Q1). The labels under the system names show the average number of requests and the mean time between the requests (calculated by taking the average of the average request intervals per task) for the three runs and all tasks.

systems deteriorated with the increase of the error rate. However, an interesting observation is that the errors had different impact on different systems reflecting the different interactive strategies employed by the systems. In some cases, erroneous answers from the oracle had the highest impact on the recall, in other cases on the precision, and in others still both measures were significantly affected. Also interesting is the fact that the impact of the errors was linear in some systems and supralinear in others, as reflected by their performance in relation to the oracle's alignment. A supralinear impact of the errors indicates that the system is making inferences from the user and thus deciding on the classification of multiple correspondence candidates based on user feedback about only one correspondence. This is an effective strategy for reducing the burden on the user, but alas leaves the matching system more susceptible to user errors. An extreme example of this is JarvisOM on the Anatomy dataset, as it uses an active-learning approach based on solely 7 user requests, and consequently is profoundly affected when faced with user errors given the size of the Anatomy dataset alignment. Curiously, this system behaves very differently in the Conference dataset, showing a linear impact of the errors, as in this case 7 requests (which is the average number it makes per task)

represent a much more substantial portion of the Conference alignments (50%) and thus leads to less inferences and consequently less impact of errors.

Apart from JarvisOM, all the systems make use of user interactions exclusively in post-matching steps to filter their candidate correspondences. LogMap and AML both request feedback on only selected correspondence candidates (based on their similarity patterns or their involvement in unsatisfiabilities). By contrast, ServOMBI employs the user to validate all its correspondence candidates (after two distinct matching stages), which corresponds to user validation rather than interactive matching. Consequently, it makes a much greater number of user requests than the other systems, and in being the system most dependent on the user, is also the one most affected by user errors.

With regard still to the number of user requests, it is interesting to note that both ServOMBI and LogMap generally increased the number of requests with the error, whereas AML and JarvisOM kept their number approximately constant. The increase is natural, as user errors can lead to more complex decision trees when interaction is used in filtering steps and inferences are drawn from the user feedback (such as during alignment repair) which leads to an increased number of subsequent requests. JarvisOM is not affected by this because it uses interaction during matching and makes a fixed 7-8 requests per matching task, whereas AML prevents it by employing a maximum query limit and stringent stopping criteria.

Two models for system response times are frequently used in the literature [7]: Shneiderman and Seow take different approaches to categorize the response times. Shneiderman takes task-centered view and sort out the response times in four categories according to task complexity: typing, mouse movement (50-150 ms), simple frequent tasks (1 s), common tasks (2-4 s) and complex tasks (8-12 s). He suggests that the user is more tolerable to delays with the growing complexity of the task at hand. Unfortunately no clear definition is given for how to define the task complexity. The Seow's model looks at the problem from a user-centered perspective by considering the user expectations towards the execution of a task: instantaneous (100-200 ms), immediate (0.5-1 s), continuous (2-5 s), captive (7-10 s); Ontology matching is a cognitively demanding task and can fall into the third or forth categories in both models. In this regard the response times (request intervals as we call them above) observed with the Anatomy dataset (with the exception of several measurements for ServOMBI) fall into the tolerable and acceptable response times in both models. The same applies for the average requests intervals for the 6 tasks in the largebio dataset. The average request intervals for the Conference dataset are lower (with the exception of ServOMBI) than those discussed for the Anatomy dataset. It could be the case however that the user could not take advantage of very low response times because the task complexity may result in higher user response time (analogically it measures the time the user needs to respond to the system after the system is ready).

9 Ontology Alignment For Query Answering (OA4QA)

Ontology matching systems rely on lexical and structural heuristics and the integration of the input ontologies and the alignments may lead to many undesired logical consequences. In [21], three principles were proposed to minimize the number of potentially

unintended consequences, namely: *(i) consistency principle*, the alignment should not lead to unsatisfiable classes in the integrated ontology; *(ii) locality principle*, the correspondences should link entities that have similar *neighborhoods*; *(iii) conservativity principle*, the alignments should not introduce alterations in the classification of the input ontologies. The occurrence of these violations is frequent, even in the reference alignments sets of the Ontology Alignment Evaluation Initiative (OAEI) [35, 36].

Violations to these principles may hinder the usefulness of ontology matching. The practical effect of these violations, however, is clearly evident when ontology alignments are involved in complex tasks such as query answering [26]. The traditional tracks of OAEI evaluate ontology matching systems w.r.t. scalability, multi-lingual support, instance matching, reuse of background knowledge, etc. Systems' effectiveness is, however, only assessed by means of classical information retrieval metrics, i.e., precision, recall and F-measure, w.r.t. a manually-curated reference alignment, provided by the organizers. The OA4QA track [37], introduced in 2015, evaluates these same metrics, with respect to the ability of the generated alignments to enable the answer of a set of queries in an ontology-based data access (OBDA) scenario, where several ontologies exist. Our target scenario is an OBDA scenario where one ontology provides the vocabulary to formulate the queries (QF-Ontology) and the second is linked to the data and it is not visible to the users (DB-Ontology). Such OBDA scenario is presented in real-world use cases, e.g., the Optique project[15] [19, 24, 35]. The integration via ontology alignment is required since only the vocabulary of the DB-Ontology is connected to the data. OA4QA will also be key for investigating the effects of logical violations affecting the computed alignments, and evaluating the effectiveness of the repair strategies employed by the matchers.

9.1 Dataset

The set of ontologies coincides with that of the conference track (§5), in order to facilitate the understanding of the queries and query results. The dataset is however extended with synthetic ABoxes, extracted from the *DBLP* dataset.[16]

Given a query q expressed using the vocabulary of ontology $\mathcal{O}_1$, another ontology $\mathcal{O}_2$ enriched with synthetic data is chosen. Finally, the query is executed over the aligned ontology $\mathcal{O}_1 \cup \mathcal{M} \cup \mathcal{O}_2$, where $\mathcal{M}$ is an alignment between $\mathcal{O}_1$ and $\mathcal{O}_2$. Here $\mathcal{O}_1$ plays the role of QF-Ontology, while $\mathcal{O}_2$ that of DB-Ontology.

9.2 Query evaluation engine

The considered evaluation engine is an extension of the OWL 2 reasoner HermiT, known as OWL-BGP[17] [25]. OWL-BGP is able to process SPARQL queries in the SPARQL-OWL fragment, under the OWL 2 Direct Semantics entailment regime [25]. The queries employed in the OA4QA track are standard conjunctive queries, that are fully supported by the more expressive SPARQL-OWL fragment. SPARQL-OWL, for instance, also

[15] http://www.optique-project.eu/
[16] http://dblp.uni-trier.de/xml/
[17] https://code.google.com/p/owl-bgp/

support queries where variables occur within complex class expressions or bind to class or property names.

9.3 Evaluation metrics and gold standard

The evaluation metrics used for the OA4QA track are the classic information retrieval ones, i.e., precision, recall and F-measure, but on the result set of the query evaluation. In order to compute the gold standard for query results, the publicly available reference alignments *ra1* has been manually revised. The aforementioned metrics are then evaluated, for each alignment computed by the different matching tools, against the *ra1*, and manually repaired version of *ra1* from conservativity and consistency violations, called *rar1* (not to be confused with *ra2* alignment of the conference track).

Three categories of queries are considered in OA4QA: *(i)* basic queries: instance retrieval queries for a single class or queries involving at most one trivial correspondence (that is, correspondences between entities with (quasi-)identical names), *(ii)* queries involving (consistency or conservativity) violations, *(iii)* advanced queries involving nontrivial correspondences.

For unsatisfiable ontologies, we tried to apply an additional repair step, that consisted in the removal of all the individuals of incoherent classes. In some cases, this allowed to answer the query, and depending on the classes involved in the query itself, sometimes it did not interfere in the query answering process.

9.4 Impact of the mappings in the query results

The impact of unsatisfiable ontologies, related to the consistency principle, is immediate. The conservativity principle, compared to the consistency principle, received less attention in literature, and its effects in a query answering process is probably less known. For instance, consider the aligned ontology $\mathcal{O}_{\mathcal{U}}$ computed using *confof* and *ekaw* as input ontologies ($\mathcal{O}_{confof}$ and $\mathcal{O}_{ekaw}$, respectively), and the *ra1* reference alignment between them. $\mathcal{O}_{\mathcal{U}}$ entails $ekaw{:}Student \sqsubseteq ekaw{:}Conf\_Participant$, while O_{ekaw} does not, and therefore this represents a conservativity principle violation [35]. Clearly, the result set for the query $q(x) \leftarrow ekaw{:}Conf\_Participant(x)$ will erroneously contain any student not actually participating at the conference. The explanation for this entailment in $\mathcal{O}_{\mathcal{U}}$ is given below, where Axioms 1 and 3 are correspondences from the reference alignment.

$$confof{:}Scholar \equiv ekaw{:}Student \tag{1}$$

$$confof{:}Scholar \sqsubseteq confof{:}Participant \tag{2}$$

$$confof{:}Participant \equiv ekaw{:}Conf\_Participant \tag{3}$$

In what follows, we provide possible (minimal) alignment repairs for the aforementioned violation:

- the weakening of Axiom 1 into $confof{:}Scholar \sqsupseteq ekaw{:}Student$,
- the weakening of Axiom 3 into $confof{:}Participant \sqsupseteq ekaw{:}Conf\_Participant$.

Repair strategies could disregard weakening in favor of complete correspondence removal, in this case the removal of either Axiom 1, or Axiom 3 could be possible repairs. Finally, for strategies including the input ontologies as a possible repair target, the removal of Axiom 2 can be proposed as a legal solution to the problem.

9.5 Results

Table 28 shows the average precision, recall and f-measure results for the whole set of queries. Matchers are evaluated on 18 queries in total, for which the sum of expected answers is 1724. Some queries have only 1 answer while other have as many as 196. AML, DKPAOM, LogMap, LogMap-C and XMap were the only matchers whose alignments allowed to answer all the queries of the evaluation.

AML was the best performing tool for what concerns averaged precision (same value as XMAP), recall (same value as LogMap) and F-measure, closely followed by LogMap, LogMap-C and XMap.

Considering Table 28, the difference in results between the publicly available reference alignment of conference track (*ra1*) and its repaired version (*rar1*, not to be confused with *ra2* of the conference track) was not significant. The F-measure ranking between the two reference alignments is almost totally preserved, the only notable variation concerns Lily, which is ranked 11th w.r.t. *ra1*, and 9th w.r.t. *rar1* (improving its results w.r.t. GMap and LogMapLt).

If we compare Table 28 (the results of the present track) and Table 6, page 14 (w.r.t. the results of conference track) we can see that 3 out of4 matchers in the top-4 ranking are shared, even if the ordering is different. Considering *rar1* alignment, the gap between the best performing matchers and the others is highlighted, and it also allows to differentiate more among the least performing matchers, and seems therefore more suitable as a reference alignment in the context of the OA4QA track evaluation.

Comparing Table 28 to Table 6 for what concerns the logical violations of the different matchers participating at the conference track, it seems that a negative correlation between the ability of answering queries and the average degree of incoherence of the matchers exists. For instance, taking into account the different positions in the ranking of LogMapLt (the version of LogMap not equipped with logical repair facilities), we can see that it is penalized more in our test case than in the traditional conference track, due to its target scenario. ServOMBI, instead, even if presenting many violations and even if most of its alignment is suffering from incoherences, is in general able to answer enough of the test queries (6 out of 18).

LogMapC, to the best of our knowledge the only ontology matching systems fully addressing conservativity principle violations, did not outperform LogMap, because some correspondences removed by its extended repair capabilities prevented to answer one of the queries (the result set was empty as an effect of correspondence removal).

9.6 Conclusions

Alignment repair does not only affect precision and recall while comparing the computed alignment w.r.t. a reference alignment, but it can enable or prevent the capability

Table 28. OA4QA track, averaged precision and recall (over the single queries), for each matcher. F-measure, instead, is computed using the averaged precision and recall. Matchers are sorted on their F-measure values for *ra1*.

Matcher	Answered queries	ra1			rar1		
		Prec.	F-m.	Rec.	Prec.	F-m.	Rec.
AML	18/18	0.78	0.76	0.75	0.76	0.75	0.75
LogMap	18/18	0.75	0.75	0.75	0.73	0.73	0.73
XMAP	18/18	0.78	0.72	0.68	0.72	0.70	0.67
LogMapC	18/18	0.72	0.71	0.69	0.72	0.71	0.70
COMMAND	14/18	0.72	0.66	0.61	0.69	0.62	0.56
DKPAOM	18/18	0.67	0.64	0.62	0.67	0.66	0.65
Mamba	14/18	0.71	0.61	0.53	0.71	0.61	0.54
CroMatcher	12/18	0.70	0.57	0.48	0.61	0.49	0.4
LogMapLt	11/18	0.70	0.52	0.42	0.58	0.43	0.35
GMap	9/18	0.65	0.49	0.39	0.61	0.43	0.33
Lily	11/18	0.64	0.47	0.37	0.64	0.48	0.39
JarvisOM	17/18	0.43	0.43	0.43	0.43	0.41	0.39
ServOMBI	6/18	0.67	0.33	0.22	0.67	0.33	0.22
RSDLWB	6/18	0.39	0.25	0.18	0.39	0.19	0.13

of an alignment to be used in a query answering scenario. As experimented in the evaluation, the conservativity violations repair technique of LogMapC on one hand improved its performances on some queries w.r.t. LogMap matcher, but in one cases it actually prevented to answer a query due to a missing correspondence. This conflicting effect in the process of query answering imposes a deeper reflection on the role of ontology alignment debugging strategies, depending on the target scenario, similarly to what already discussed in [30] for incoherence alignment debugging.

The results we presented depend on the considered set of queries. What clearly emerges is that the role of logical violations is playing a major role in our evaluation, and a possible bias due to the set of chosen queries can be mitigated by an extended set of queries and synthetic data. We hope that this will be useful in the further exploration of the findings of this first edition of the OA4QA track.

As a final remark, we would like to clarify that the entailment of new knowledge, obtained using the alignments, is not always negative, and conservativity principle violations can be false positives. Another extension to the current set of queries would target such false positives, with the aim of penalizing the indiscriminate repairs in presence of conservativity principle violations.

10 Instance matching

The instance matching track aims at evaluating the performance of matching tools identify relations between pairs of items/instances found in Aboxes. The track is organized in five independent tasks, namely *author disambiguation* (*author-dis task*), *author recognition* (*author-rec task*), *value semantics* (*val-sem task*), *value structure* (*val-struct task*), and *value structure semantics* (*val-struct-sem task*).

Each task is articulated in two tests, namely *sandbox* and *mainbox*, with different scales, i.e., number of instances to match:

- *Sandbox* (small scale) is an open test, meaning that the set of expected mappings, i.e., reference alignment, is given in advance to the participants.
- *Mainbox* (medium scale) is a blind test, meaning that the reference alignment is not given in advance to the participants.

Each test contains two datasets called source and target and the goal is to discover the matching pairs, i.e., mappings, among the instances in the source dataset and the instances in the target dataset.

For the sake of clarity, we split the presentation of task results in two different sections as follows.

10.1 Results for author disambiguation (author-dis) and author recognition (author-rec) tasks

The goal of author-dis and author-rec tasks is to discover links between pairs of OWL instances referring to the same person, i.e., author, based on their publications. In both tasks, expected mappings are 1:1 (one person of the source dataset corresponds to exactly one person of the target dataset and vice versa).

About the author-dis task, in both source and target datasets, authors and publications are described as instances of the classes `http://islab.di.unimi.it/imoaei2015#Person` and `http://islab.di.unimi.it/imoaei2015#Publication`, respectively. Publications are associated with the corresponding person instance through the property `http://islab.di.unimi.it/imoaei2015#author_of`. Author and publication information are differently described in the two datasets. For example, only the first letter of author names and the initial part of publication titles are shown in the target dataset while the full strings are provided in the source datasets. The matching challenge regards the capability to resolve such a kind of ambiguities on author and publication descriptions.

About the author-rec task, author and publication descriptions in the source dataset are analogous to those in the author-dis task. As a difference, in the target dataset, each author/person is only associated with a publication titled "Publication report" containing aggregated information, such as number of publications, h-index, years of activity, and number of citations. The matching challenge regards the capability to link a person in the source dataset with the person in the target dataset containing the corresponding publication report.

Participants to author-dis and author-rec tasks are EXONA, InsMT+, Lily, LogMap, and RiMOM. Results are shown in Table 29 and 30, respectively.

For each tool, we provide the number of mapping expected in the ground truth, the number of mapping actually retrieved by the tool, and tool performances in terms of precision, recall, and F-measure.

On the author-dis task, we note that good results in terms of precision and recall are provided by all the participating tools. As a general remark, precision values are slightly better than recall values. This behavior highlights the consolidated maturity of

	Exp. mappings	Retr. mappings	Prec.	F-m.	Rec.
Sandbox task					
EXONA	854	854	0.94	0.94	0.94
InsMT+	854	722	0.83	0.76	0.70
Lily	854	854	0.98	0.98	0.98
LogMap	854	779	0.99	0.95	0.91
RiMOM	854	854	0.93	0.93	0.93
Mainbox task					
EXONA	8428	144827	0.0	NaN	0.0
InsMT+	8428	7372	0.76	0.71	0.66
Lily	8428	8428	0.96	0.96	0.96
LogMap	7030	779	0.99*	0.91	0.83
RiMOM	8428	8428	0.91	0.91	0.91

Table 29. Results of the author-dis task (.99* should have been rounded to 1.0).

	Exp. mappings	Retr. mappings	Prec.	F-m.	Rec.
Sandbox task					
EXONA	854	854	0.52	0.52	0.52
InsMT+	854	90	0.56	0.11	0.06
Lily	854	854	1.0	1.0	1.0
LogMap	854	854	1.0	1.0	1.0
RiMOM	854	854	1.0	1.0	1.0
Mainbox task					
EXONA	8428	8428	0.41	0.41	0.41
InsMT+	8428	961	0.25	0.05	0.03
Lily	8428	8424	0.99*	0.99*	0.99*
LogMap	8436	779	0.99*	0.99*	1.0
RiMOM	8428	8428	0.99*	0.99*	0.99*

Table 30. Results of the author-rec task (.99* should have been rounded to 1.0).

instance matching tools when the alignment goal is to handle syntax modifications in instance descriptions. On the author-rec task, the differences in tool performances are more marked. In particular, we note that Lily, LogMap, and RiMOM have better results than EXONA and InsMT+. Probably, this is due to the fact that the capability to align the summary publication report to the appropriate author requires reasoning functionalities that are available to only a subset of the participating tools. The distinction between sandbox and mainbox tests puts in evidence that the capability to handle large-scale datasets is complicated for most of the participating tools. We note that LogMap and RiMOM are the best performing tools on the mainbox tests, but very-long execution times usually characterize participants in the execution of large-scale tests. We argue that this is a forthcoming challenging issue in the field of instance matching, on which further experimentations and tests need to focus in the future competitions.

10.2 Results for value semantics (val-sem), value structure (val-struct), and value structure semantics (val-struct-sem) tasks

The val-sem, val-struct, and val-struct-sem tasks are three evaluation tasks of instance matching tools where the goal is to determine when two OWL instances describe the same real world object. The datasets have been produced by altering a set of source data and generated by SPIMBENCH [32] with the aim to generate descriptions of the same entity where value-based, structure-based and semantics-aware transformations are employed in order to create the target data. The value-based transformations consider mainly typographical errors and different data formats, the structure-based transformations consider transformations applied on the structure of object and datatype properties and the semantics-aware transformations are transformations at the instance level considering the schema. The latter are used to examine if the matching systems take into account RDFS and OWL constructs in order to discover correspondences between instances that can be found only by considering schema information.

We stress that an instance in the source dataset can have none or one matching counterpart in the target dataset. A dataset is composed of a Tbox and a corresponding Abox. Source and target datasets share almost the same Tbox (with some difference in the properties' level, due to the structure-based transformations). Ontology is described through 22 classes, 31 datatype properties, and 85 object properties. From those properties, there is 1 an inverse functional property and 2 are functional properties. The sandbox scale is 10K instances while the mainbox scale is 100K instances.

We asked the participants to match the Creative Works instances (NewsItem, BlogPost and Programme) in the source dataset against the instances of the corresponding class in the target dataset. We expected to receive a set of links denoting the pairs of matching instances that they found to refer to the same entity. The datasets of the val-sem task have been produced by altering a set of source data through value-based and semantics-aware transformations, while val-struct through value-based and structure-based transformations and val-struct-sem task through value-based, structure-based and semantics-aware.

The participants to these tasks are LogMap and STRIM. For evaluation, we built a ground truth containing the set of expected links where an instance i_1 in the source dataset is associated with an instance in the target dataset that has been generated as an altered description of i_1.

The way that the transformations were done, was to apply value-based, structure-based and semantics-aware transformations, on different triples pertaining to one class instance. For example, regarding the val-struct task, for an instance u_1, we performed a value-based transformation on its triple (u_1, p_1, o_1) where p_1 is a data type property and a structure-based transformation on its triple (u_1, p_2, o_2).

The evaluation has been performed by calculating precision, recall, and F-measure and results are provided in Tables 31, 32, 33.

The main comment is that the quality of the results for both LogMap and STRIM is very high as we created the tasks val-sem, val-struct, and val-struct-sem in order to be the easiest ones. LogMap and STRIM have consistent behavior for the sandbox and the mainbox tasks, a fact that shows that both systems can handle different sizes of data without reducing their performance.

LogMap's performance drops for tasks that consider structure-based transformations (val-struct and val-struct-sem). Also, it produces links that are quite often correct (resulting in a good precision) but fails in capturing a large number of the expected links (resulting in a lower recall). STRIM's performance drops for tasks that consider semantics-aware transformations (val-sem and val-struct-sem) as expected. The probability of capturing a correct link is high, but the probability of a retrieved link to be correct is lower, resulting in a high recall but not equally high precision.

	Exp. mappings	Retr. mappings	Prec.	F-m.	Rec.
Sandbox task					
STRIM	9649	10641	0.91	0.95	0.99*
LogMap	9649	8350	0.99	0.92	0.86
Mainbox task					
STRIM	97256	106232	0.91	0.95	0.99*
LogMap	97256	83880	0.99*	0.92	0.86

Table 31. Results of the value-semantics task (.99* should have been rounded to 1.0).

	Exp. mappings	Retr. mappings	Prec.	F-m.	Rec.
Sandbox task					
STRIM	10601	10657	0.99	0.99*	0.99*
LogMap	10601	8779	0.99	0.90	0.82
Mainbox task					
STRIM	106137	105352	0.99	0.99	0.99*
LogMap	106137	87137	0.99*	0.90	0.82

Table 32. Results of the value-structure task (.99* should have been rounded to 1.0).

	Exp. mappings	Retr. mappings	Prec.	F-m.	Rec.
Sandbox task					
STRIM	9790	10639	0.92	0.96	0.99*
LogMap	9790	7779	0.99	0.88	0.79
Mainbox task					
STRIM	98144	106576	0.92	0.95	0.99*
LogMap	98144	77983	0.99*	0.88	0.79

Table 33. Results of the value-structure-semantics task (.99* should have been rounded to 1.0).

11 Lesson learned and suggestions

Here are lessons learned from running OAEI 2015:

A) This year indicated again that requiring participants to implement a minimal interface was not a strong obstacle to participation with some exceptions. Moreover, the community seems to get used to the SEALS infrastructure introduced for OAEI 2011.
B) It would be useful to tighten the rules for evaluation so that the we can again write that "All tests have been run entirely from the SEALS platform with the strict same protocol" and we do not end up with one evaluation setting tailored for each system. This does not mean that we should come back to the exact setting of two years ago, but that evaluators and tool developers should decide for one setting and stick to it (i.e. avoid system variants participating only in a concrete track).
C) This year, thanks to Daniel Faria, we updated the SEALS client to include the new functionalities introduced in the interactive matching track. We also updated the client to use the latest libraries which caused some trouble to some Jena developers.
D) This year, due to technical problems, we were missing the SEALS web portal, but this dis not seem to affect the participation since the number of submitted systems increased with respect to 2014. In any case, we hope to bring back the SEALS portal for future OAEI campaigns.
E) As already proposed in previous years, it would be good to set the preliminary evaluation results by the end of July to avoid last minute errors and incompatibilities with the SEALS client.
F) Again, given the high number of publications on data interlinking, it is surprising to have so few participants to the instance matching track, although this number has increased. Nevertheless, we are in direct contact with data interlinking system developers that may be interested in integrating their benchmarks within the OAEI.
G) As in previous years we had a panel discussion session during the OM workshop where we discussed about hot topics and future lines for the OAEI. Among others, we discussed about the need of continuing the effort of improving the interactive track and adding uncertainty to the OAEI benchmarks (as in the Conference track). Furthermore we also analyzed the feasibility of joining efforts with the *Process Model Matching Contest (PMMC)*: `https://ai.wu.ac.at/emisa2015/contest.php`. As a first step we planned to make available an interface to convert from/to a model specification to OWL in order to ease the participation of OAEI systems in the PMMC and vice versa.

Here are lessons learned per OAEI 2015 track:

A) Most of the systems participating in the Multifarm track pre-compiles a local dictionary in order to avoid multiple accesses to the translators within the matching process which would exceed the allowed (free) translation quota. For future years we may consider limiting the amount of local information a system can store.
B) In order to attract more instance matching systems to participate in value semantics (val-sem), value structure (val-struct), and value structure semantics (val-struct-sem) tasks, we need to produce benchmarks that have fewer instances (in the order

of 10000), of the same type (in our benchmark we asked systems to compare instances of different types). To balance those aspects, we must then produce benchmarks that are more complex i.e., contain more complex transformations.

C) In the largebio track we flagged incoherence-causing mappings (i.e., those removed by at least one of the used repair approaches: Alcomo [26], LogMap [20] or AML [31]) by setting their relation to "?" (unknown). These "?" mappings are neither considered as positive nor as negative when evaluating the participating ontology matching systems, but will simply be ignored. The interactive track uses the reference alignments of each track to simulate the user interaction or Oracle. This year, when simulating the user interaction with the largebio dataset, the Oracle returned "true" when asked about a mapping flagged as "unknown". However, we realized that returning true leads to erratic behavior (and loss of performance) for algorithms computing an interactive repair. Thus, as the role of user feedback during repair is extremely important, we should ensure that the Oracle's behavior simulates it in a sensible manner.

D) Based on the uncertain reference alignment from the conference track we conclude that many more matchers provide alignments with a range of confidence values than in the past which better corresponds to human evaluation of the match quality.

E) In the interactive track we simulate users with different error rates, i.e., given a query about a mapping there is a random chance that the user is wrong. A "smart" interactive system could potentially ask the same question several times in order to mitigate the effect of the simulated error rate of the user. In the future we plan to extend the SEALS client to identify this potential behavior in interactive matching systems.

F) For the OA4QA track, both averaging F-measures and computing it from the averaged precision and recall values raised confusion while reporting the results. For the next edition we plan to use a global precision and recall (and consequently F-measure) on the combined result sets of all the query, similarly to what is already done in the conference track. One major challenge in the design of the new scoring function is to keep the scoring balanced despite differences in cardinality of the result sets of the single queries.

12 Conclusions

OAEI 2015 saw an increased number of participants. We hope to keep this trend next year. Most of the test cases are performed on the SEALS platform, including the instance matching track. This is good news for the interoperability of matching systems. The fact that the SEALS platform can be used for such a variety of tasks is also a good sign of its relevance.

Again, we observed improvements of runtimes. For example, all systems but two participating in the anatomy track finished in less than 15 minutes. As usual, most of the systems favor precision over recall. In general, participating matching systems do not take advantage of alignment repairing system and return sometimes incoherent alignments. This is a problem if their result has to be taken as input by a reasoning system.

This year we also evaluated ontology matching systems in query answering tasks. The track was not fully based on SEALS but it reused the computed alignments from

the conference track, which runs in the SEALS client. This new track shed light on the performance of ontology matching systems with respect to the coherence of their computed alignments.

A novelty of this year was an extended evaluation in the conference, interactive and instance matching tracks. This brought interesting insights on the performances of such systems and should certainly be continued.

Most of the 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. Matching evaluation still remains a challenging topic, which is worth further research in order to facilitate the progress of the field [33]. More 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 useful reports on their experience. The best way to learn about the results remains to read the following papers.

We are very grateful to the Universidad Politécnica de Madrid (UPM), especially to Nandana Mihindukulasooriya and Asunción Gómez Pérez, for moving, setting up and providing the necessary infrastructure to run the SEALS repositories.

We are also 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 thank Christian Meilicke for his support of the anatomy test case.

We thank Khiat Abderrahmane for his support in the Arabic data set and Catherine Comparot for her feedback and support in the MultiFarm test case.

We also thank for their support 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), Heiner Stuckenschmidt (Mannheim Universität, DE), George Vouros (University of the Aegean, GR).

Jérôme Euzenat, Ernesto Jimenez-Ruiz, and Cássia Trojahn dos Santos have been partially supported by the SEALS (IST-2009-238975) European project in the previous years.

Ernesto has also been partially supported by the Seventh Framework Program (FP7) of the European Commission under Grant Agreement 318338, “Optique”, the Royal Society, and the EPSRC projects Score!, DBOnto and MaSI^3.

Ondřej Zamazal has been supported by the CSF grant no. 14-14076P.

Daniel Faria was supported by the Portuguese FCT through the SOMER project (PTDC/EIA-EIA/119119/2010), and the LASIGE Strategic Project (PEst-OE/EEI/UI0408/ 2015).

Michelle Cheatham has been supported by the National Science Foundation award ICER-1440202 “EarthCube Building Blocks: Collaborative Proposal: GeoLink”.

References

1. José Luis Aguirre, Bernardo Cuenca Grau, Kai Eckert, Jérôme Euzenat, Alfio Ferrara, Robert Willem van Hague, Laura Hollink, Ernesto Jiménez-Ruiz, Christian Meilicke, Andriy Nikolov, Dominique Ritze, François Scharffe, Pavel Shvaiko, Ondrej Sváb-Zamazal, Cássia Trojahn, and Benjamin Zapilko. Results of the ontology alignment evaluation initiative 2012. In *Proc. 7th ISWC ontology matching workshop (OM), Boston (MA US)*, pages 73–115, 2012.
2. Benhamin Ashpole, Marc Ehrig, Jérôme Euzenat, and Heiner Stuckenschmidt, editors. *Proc. K-Cap Workshop on Integrating Ontologies*, Banff (Canada), 2005.
3. Olivier Bodenreider. The unified medical language system (UMLS): integrating biomedical terminology. *Nucleic Acids Research*, 32:267–270, 2004.
4. 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 ISWC ontology matching workshop (OM), Karlsruhe (DE)*, pages 73–120, 2008.
5. Michelle Cheatham and Pascal Hitzler. Conference v2. 0: An uncertain version of the oaei conference benchmark. In *The Semantic Web–ISWC 2014*, pages 33–48. Springer, 2014.
6. Bernardo Cuenca Grau, Zlatan Dragisic, Kai Eckert, Jérôme Euzenat, Alfio Ferrara, Roger Granada, Valentina Ivanova, Ernesto Jiménez-Ruiz, Andreas Oskar Kempf, Patrick Lambrix, Andriy Nikolov, Heiko Paulheim, Dominique Ritze, François Scharffe, Pavel Shvaiko, Cássia Trojahn dos Santos, and Ondrej Zamazal. Results of the ontology alignment evaluation initiative 2013. In Pavel Shvaiko, Jérôme Euzenat, Kavitha Srinivas, Ming Mao, and Ernesto Jiménez-Ruiz, editors, *Proc. 8th ISWC workshop on ontology matching (OM), Sydney (NSW AU)*, pages 61–100, 2013.
7. Jim Dabrowski and Ethan V. Munson. 40 years of searching for the best computer system response time. *Interacting with Computers*, 23(5):555–564, 2011.
8. Jérôme David, Jérôme Euzenat, François Scharffe, and Cássia Trojahn dos Santos. The alignment API 4.0. *Semantic web journal*, 2(1):3–10, 2011.
9. Zlatan Dragisic, Kai Eckert, Jérôme Euzenat, Daniel Faria, Alfio Ferrara, Roger Granada, Valentina Ivanova, Ernesto Jiménez-Ruiz, Andreas Oskar Kempf, Patrick Lambrix, Stefano Montanelli, Heiko Paulheim, Dominique Ritze, Pavel Shvaiko, Alessandro Solimando, Cássia Trojahn dos Santos, Ondrej Zamazal, and Bernardo Cuenca Grau. Results of the ontology alignment evaluation initiative 2014. In *Proceedings of the 9th International Workshop on Ontology Matching collocated with the 13th International Semantic Web Conference (ISWC 2014), Riva del Garda, Trentino, Italy, October 20, 2014.*, pages 61–104, 2014.

10. 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 ISWC ontology matching workshop (OM), Chantilly (VA US)*, pages 73–126, 2009.
11. 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 *Proc. 5th ISWC ontology matching workshop (OM), Shanghai (CN)*, pages 85–117, 2010.
12. 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 *Proc. 6th ISWC ontology matching workshop (OM), Bonn (DE)*, pages 85–110, 2011.
13. 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 ISWC ontology matching workshop (OM), Busan (KR)*, pages 96–132, 2007.
14. 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.
15. 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 ISWC ontology matching workshop (OM), Athens (GA US)*, pages 73–95, 2006.
16. Jérôme Euzenat, Maria Rosoiu, and Cássia Trojahn dos Santos. Ontology matching benchmarks: generation, stability, and discriminability. *Journal of web semantics*, 21:30–48, 2013.
17. Jérôme Euzenat and Pavel Shvaiko. *Ontology matching*. Springer-Verlag, Heidelberg (DE), 2nd edition, 2013.
18. Daniel Faria, Ernesto Jiménez-Ruiz, Catia Pesquita, Emanuel Santos, and Francisco M. Couto. Towards Annotating Potential Incoherences in BioPortal Mappings. In *13th International Semantic Web Conference*, volume 8797 of *Lecture Notes in Computer Science*, pages 17–32. Springer, 2014.
19. Martin Giese, Ahmet Soylu, Guillermo Vega-Gorgojo, Arild Waaler, Peter Haase, Ernesto Jiménez-Ruiz, Davide Lanti, Martín Rezk, Guohui Xiao, Özgür L. Özçep, and Riccardo Rosati. Optique: Zooming in on big data. *IEEE Computer*, 48(3):60–67, 2015.
20. Ernesto Jiménez-Ruiz and Bernardo Cuenca Grau. LogMap: Logic-based and scalable ontology matching. In *Proc. 10th International Semantic Web Conference (ISWC), Bonn (DE)*, pages 273–288, 2011.
21. 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.
22. Ernesto Jiménez-Ruiz, Christian Meilicke, Bernardo Cuenca Grau, and Ian Horrocks. Evaluating mapping repair systems with large biomedical ontologies. In *Proc. 26th Description Logics Workshop*, 2013.
23. Yevgeny Kazakov, Markus Krötzsch, and Frantisek Simancik. Concurrent classification of EL ontologies. In *Proc. 10th International Semantic Web Conference (ISWC), Bonn (DE)*, pages 305–320, 2011.
24. Evgeny Kharlamov, Dag Hovland, Ernesto Jiménez-Ruiz, Davide Lanti, Hallstein Lie, Christoph Pinkel, Martín Rezk, Martin G. Skjæveland, Evgenij Thorstensen, Guohui Xiao,

Dmitriy Zheleznyakov, and Ian Horrocks. Ontology based access to exploration data at statoil. In *The Semantic Web - ISWC 2015 - 14th International Semantic Web Conference, Bethlehem, PA, USA, October 11-15, 2015, Proceedings, Part II*, pages 93–112, 2015.
25. Ilianna Kollia, Birte Glimm, and Ian Horrocks. SPARQL query answering over OWL ontologies. In *The Semantic Web: Research and Applications*, pages 382–396. Springer, 2011.
26. Christian Meilicke. *Alignment Incoherence in Ontology Matching*. PhD thesis, University Mannheim, 2011.
27. Christian Meilicke, Raúl García Castro, Frederico Freitas, Willem Robert van Hage, Elena Montiel-Ponsoda, Ryan Ribeiro de Azevedo, Heiner Stuckenschmidt, Ondrej Sváb-Zamazal, Vojtech Svátek, Andrei Tamilin, Cássia Trojahn, and Shenghui Wang. MultiFarm: A benchmark for multilingual ontology matching. *Journal of web semantics*, 15(3):62–68, 2012.
28. Boris Motik, Rob Shearer, and Ian Horrocks. Hypertableau reasoning for description logics. *Journal of Artificial Intelligence Research*, 36:165–228, 2009.
29. Heiko Paulheim, Sven Hertling, and Dominique Ritze. Towards evaluating interactive ontology matching tools. In *Proc. 10th Extended Semantic Web Conference (ESWC), Montpellier (FR)*, pages 31–45, 2013.
30. Catia Pesquita, Daniel Faria, Emanuel Santos, and Francisco Couto. To repair or not to repair: reconciling correctness and coherence in ontology reference alignments. In *Proc. 8th ISWC ontology matching workshop (OM), Sydney (AU)*, page this volume, 2013.
31. Emanuel Santos, Daniel Faria, Catia Pesquita, and Francisco Couto. Ontology alignment repair through modularization and confidence-based heuristics. *CoRR*, abs/1307.5322, 2013.
32. Tzanina Saveta, Evangelia Daskalaki, Giorgos Flouris, Irini Fundulaki, Melanie Herschel, and Axel-Cyrille Ngonga Ngomo. Pushing the limits of instance matching systems: A semantics-aware benchmark for linked data. In *WWW, Companion Volume*, 2015.
33. Pavel Shvaiko and Jérôme Euzenat. Ontology matching: state of the art and future challenges. *IEEE Transactions on Knowledge and Data Engineering*, 25(1):158–176, 2013.
34. Alessandro Solimando, Ernesto Jiménez-Ruiz, and Giovanna Guerrini. Detecting and correcting conservativity principle violations in ontology-to-ontology mappings. In *The Semantic Web–ISWC 2014*, pages 1–16. Springer, 2014.
35. Alessandro Solimando, Ernesto Jiménez-Ruiz, and Giovanna Guerrini. Detecting and Correcting Conservativity Principle Violations in Ontology-to-Ontology Mappings. In *International Semantic Web Conference*, 2014.
36. Alessandro Solimando, Ernesto Jiménez-Ruiz, and Giovanna Guerrini. A multi-strategy approach for detecting and correcting conservativity principle violations in ontology alignments. In *Proceedings of the 11th International Workshop on OWL: Experiences and Directions (OWLED 2014) co-located with 13th International Semantic Web Conference on (ISWC 2014), Riva del Garda, Italy, October 17-18, 2014.*, pages 13–24, 2014.
37. Alessandro Solimando, Ernesto Jiménez-Ruiz, and Christoph Pinkel. Evaluating ontology alignment systems in query answering tasks. In *Proceedings of the ISWC 2014 Posters & Demonstrations Track a track within the 13th International Semantic Web Conference, ISWC 2014, Riva del Garda, Italy, October 21, 2014.*, pages 301–304, 2014.
38. York Sure, Oscar Corcho, Jérôme Euzenat, and Todd Hughes, editors. *Proc. ISWC Workshop on Evaluation of Ontology-based Tools (EON), Hiroshima (JP)*, 2004.

Dayton, Linköping, Grenoble, Lisbon, Milano,
Heraklion, Toulouse, Oxford, Trento, Paris, Prague
December 2015