Proceedings of The Tenth International Workshop on Ontology Matching (OM-2015)

Pavel Shvaiko, Jérôme Euzenat, Ernesto Jiménez-Ruiz, Michelle Cheatham, Oktie Hassanzadeh

Ontology Matching

OM-2015

Proceedings of the ISWC Workshop

Introduction

Ontology matching[1] is a key interoperability enabler for the semantic web, as well as a useful tactic in some classical data integration tasks dealing with the semantic heterogeneity problem. It takes ontologies as input and determines as output an alignment, that is, a set of correspondences between the semantically related entities of those ontologies. These correspondences can be used for various tasks, such as ontology merging, data translation, query answering or navigation on the web of data. Thus, matching ontologies enables the knowledge and data expressed in the matched ontologies to interoperate.

The workshop has three goals:

- To bring together leaders from *academia*, *industry* and *user institutions* to assess how academic advances are addressing real-world requirements. The workshop strives to improve academic awareness of industrial and final user needs, and therefore direct research towards those needs. Simultaneously, the workshop serves to inform industry and user representatives about existing research efforts that may meet their requirements. The workshop also investigated how the ontology matching technology is going to evolve.
- To conduct an extensive and rigorous evaluation of ontology matching and instance matching (link discovery) approaches through the OAEI (Ontology Alignment Evaluation Initiative) 2015 campaign[2]. Besides specific real-world matching tasks such as the one involving large biomedical ontologies, OAEI-2015 introduced linked data benchmarks. Therefore, the ontology matching evaluation initiative itself provided a solid ground for discussion of how well the current approaches are meeting business needs.
- To examine new uses, similarities and differences from database schema matching, which has received decades of attention but is just beginning to transition to mainstream tools.

The program committee selected 3 long and 5 short submissions for oral presentation and 9 submissions for poster presentation. 22 matching systems participated in this year's OAEI campaign. Further information about the Ontology Matching workshop can be found at: `http://om2015.ontologymatching.org/`.

[1] `http://www.ontologymatching.org/`
[2] `http://oaei.ontologymatching.org/2015`

Acknowledgments. We thank all members of the program committee, authors and local organizers for their efforts. We appreciate support from the Trentino as a Lab (TasLab)[3] initiative of the European Network of the Living Labs[4] at Informatica Trentina SpA[5], the EU SEALS (Semantic Evaluation at Large Scale)[6] project and the Semantic Valley[7] initiative.

Pavel Shvaiko
Jérôme Euzenat
Ernesto Jiménez-Ruiz
Michelle Cheatham
Oktie Hassanzadeh

October 2015

[3] http://www.taslab.eu
[4] http://www.openlivinglabs.eu
[5] http://www.infotn.it
[6] http://www.seals-project.eu
[7] http://www.semanticvalley.org/index_eng.htm

Organization

Organizing Committee

Pavel Shvaiko, Informatica Trentina SpA, Italy
Jérôme Euzenat, INRIA & University Grenoble Alpes, France
Ernesto Jiménez-Ruiz, University of Oxford, UK
Michelle Cheatham, Wright State University, USA
Oktie Hassanzadeh, IBM Research, USA

Program Committee

Alsayed Algergawy, Jena University, Germany
Michele Barbera, Spazio Dati, Italy
Zohra Bellahsene, LRIMM, France
Olivier Bodenreider, National Library of Medicine, USA
Marco Combetto, Informatica Trentina, Italy
Valerie Cross, Miami University, USA
Isabel Cruz, The University of Illinois at Chicago, USA
Jérôme David, University Grenoble Alpes & INRIA, France
Warith Eddine Djeddi, LIPAH & LABGED, Tunisia
Alfio Ferrara, University of Milan, Italy
Fausto Giunchiglia, University of Trento, Italy
Wei Hu, Nanjing University, China
Ryutaro Ichise, National Institute of Informatics, Japan
Antoine Isaac, Vrije Universiteit Amsterdam & Europeana, Netherlands
Daniel Faria, Instituto Gulbenkian de Ciência, Portugal
Patrick Lambrix, Linköpings Universitet, Sweden
Nico Lavarini, Expert System, Italy
Vincenzo Maltese, University of Trento, Italy
Robert Meusel, University of Mannheim, Germany
Fiona McNeill, University of Edinburgh, UK
Christian Meilicke, University of Mannheim, Germany
Peter Mork, Noblis, USA
Andriy Nikolov, Open University, UK
Axel Ngonga, University of Leipzig, Germany
Leo Obrst, The MITRE Corporation, USA
Heiko Paulheim, University of Mannheim, Germany
Andrea Perego, European Commission - Joint Research Centre, Italy
Catia Pesquita, University of Lisbon, Portugal
Dominique Ritze, University of Mannheim, Germany
Alessandro Solimando, University of Genova, Italy
Kavitha Srinivas, IBM, USA

Umberto Straccia, ISTI-C.N.R., Italy
Ondřej Svab-Zamazal, Prague University of Economics, Czech Republic
Cássia Trojahn, IRIT, France
Lorenzino Vaccari, European Commission - Joint Research Center, Italy
Ludger van Elst, DFKI, Germany
Shenghui Wang, Vrije Universiteit Amsterdam, Netherlands
Songmao Zhang, Chinese Academy of Sciences, China

Table of Contents

Long Technical Papers

New paradigm for alignment extraction
Christian Meilicke, Heiner Stuckenschmidt 1

A multilingual ontology matcher
Gábor Bella, Fausto Giunchiglia, Ahmed AbuRa'edy, Fiona McNeill 13

Understanding a large corpus of web tables
through matching with knowledge bases: an empirical study
Oktie Hassanzadeh, Michael J. Ward,
Mariano Rodriguez-Muro, Kavitha Srinivas 25

Short Technical Papers

Combining sum-product network and noisy-or model
for ontology matching
Weizhuo Li 35

Towards combining ontology matchers via anomaly detection
Alexander C. Müller, Heiko Paulheim 40

User involvement in ontology Matching
using an online active learning approach
Booma S. Balasubramani, Aynaz Taheri, Isabel F. Cruz 45

ADOM: arabic dataset for evaluating arabic and
cross-lingual ontology alignment systems
Abderrahmane Khiat, Moussa Benaissa, Ernesto Jiménez-Ruiz 50

Ontology matching for big data applications
in the smart dairy farming domain
Jack P.C. Verhoosel, Michael van Bekkum, Frits K. van Evert 55

OAEI Papers

Results of the Ontology Alignment Evaluation Initiative 2015
Michelle Cheatham, Zlatan Dragisic, Jérôme Euzenat, Daniel Faria, Alfio Ferrara, Giorgos Flouris, Irini Fundulaki, Roger Granada, Valentina Ivanova, Ernesto Jiménez-Ruiz, Patrick Lambrix, Stefano Montanelli, Catia Pesquita, Tzanina Saveta, Pavel Shvaiko, Alessandro Solimando, Cássia Trojahn, Ondřej Zamazal ... 60

AML results for OAEI 2015
Daniel Faria, Catarina Martins, Amruta Nanavaty, Daniela Oliveira, Booma Sowkarthiga, Aynaz Taheri, Catia Pesquita, Francisco Couto, Isabel Cruz ... 116

CLONA results for OAEI 2015
Mariem El Abdi, Hazem Souid, Marouen Kachroudi, Sadok Ben Yahia ... 124

CroMatcher results for OAEI 2015
Marko Gulić, Boris Vrdoljak, Marko Banek ... 130

DKP-AOM: results for OAEI 2015
Muhammad Fahad ... 136

EXONA results for OAEI 2015
Syrine Damak, Hazem Souid, Marouen Kachroudi, Sami Zghal ... 145

GMap: results for OAEI 2015
Weizhuo Li, Qilin Sun ... 150

InsMT+ results for OAEI 2015 instance matching
Abderrahmane Khiat, Moussa Benaissa ... 158

Lily results for OAEI 2015
Wenyu Wang, Peng Wang ... 162

LogMap family results for OAEI 2015
Ernesto Jiménez-Ruiz-Ruiz, Bernardo Cuenca Grau, Alessandro Solimando, Valerie Cross ... 171

LYAM++ results for OAEI 2015
Abdel Nasser Tigrine, Zohra Bellahsene, Konstantin Todorov ... 176

MAMBA - results for the OAEI 2015
Christian Meilicke ... 181

RiMOM results for OAEI 2015
Yan Zhang, Juanzi Li 185

RSDL workbench results for OAEI 2015
Simon Schwichtenberg, Gregor Engels 192

ServOMBI at OAEI 2015
Nouha Kheder, Gayo Diallo 200

STRIM results for OAEI 2015 instance matching evaluation
Abderrahmane Khiat, Moussa Benaissa,
Mohammed Amine Belfedhal 208

XMap: results for OAEI 2015
Warith Eddine Djeddi, Mohamed Tarek Khadir,
Sadok Ben Yahia 216

Posters

Instance-based property matching in linked open data environment
Cheng Xie, Dominique Ritze, Blerina Spahiu, Hongming Cai 222

RinsMatch: a suggestion-based instance matching system
in RDF Graphs
Mehmet Aydar, Austin Melton 224

Triple-based similarity propagation for linked data matching
Eun-Kyung Kim, Sangha Nam, Jongsung Woo,
Sejin Nam, Key-Sun Choi 226

An effective configuration learning algorithm
for entity resolution
Khai Nguyen, Ryutaro Ichise 228

Search-space reduction for post-matching
correspondence provisioning
Thomas Kowark, Hasso Plattner 230

Automatic mapping of Wikipedia categories into OpenCyc types
Aleksander Smywiński-Pohl, Krzysztof Wróbel 232

Exploiting multilinguality for ontology matching purposes
Mauro Dragoni 234

Ontology matching techniques for enterprise architecture models
Marzieh Bakhshandeh, Catia Pesquita, José Borbinha 236

MOSEW: a tool suite for service enabled work
Mostafijur Rahman, Wendy MacCaull 238

A New Paradigm for Alignment Extraction

Christian Meilicke and Heiner Stuckenschmidt

Research Group Data and Web Science
University of Mannheim, 68163 Mannheim, Germany
christian|heiner@informatik.uni-mannheim.de

Abstract. Ontology matching techniques that are based on the analysis of names usually create first a set of matching hypotheses annotated with similarity weights followed by the extraction or selection of a set of correspondences. We propose to model this last step as an optimization problem. Our proposal differs fundamentally from other approaches since both logical and linguistic entities appear as first class citizens in the optimization problem. The extraction step will not only result in a set of correspondences but will also entail assumptions related to the meaning of the tokens that appeared in the involved labels. We discuss examples that illustrate the benefits of our approach and present a Markov Logic formalization. We conduct an experimental evaluation and present first results.

1 Introduction

Ontology Matching has become a vivid field of research over the last decade. Hundreds of papers propose and discuss ontology matching techniques, introduce improvements, or present complete matching systems. Especially the system papers illustrate a general paradigm common to probably all systems using name-based alignment methods. This paradigm is the understanding of ontology matching as a sequential process that starts with analyzing different types of evidence, in most cases with a focus on the involved labels, and generates as an intermediate result a set of weighted matching hypotheses. From the intermediate result a subset of the generated hypotheses is chosen as final output. The first phase is typically dominated by the computation, aggregation, propagation, and any other method for refining similarity scores. The techniques applied in the second phase range from thresholds to the selection of coherent subsets [6, 8] that might be optimal with respect to an objective function. Most approaches model the intermediate result as a set of correspondences annotated with confidence scores. These confidence scores are aggregated values derived from an analysis of the tokens that appear in the labels of the ontological entities. With the help of several examples we argue that the extraction problem should be modeled differently such that both tokens and logical entities (classes and properties) appear as first class citizens. Otherwise it will not be possible to exploit that the acceptance or rejection of a correspondence follows from the assumption that two tokens have (or do not have) the same meaning. However, any reasonable extraction should be consistent with its underlying assumptions. This can only be ensured if the assumptions themselves can be modeled explicitly.

We presented a first sketch of this approach in [9]. Now we extend and concretize the approach including a first implementation. We present foundations in Section 2. In

Section 3 we discuss two scenarios where a classic approach makes a selection decision in an non-reasonable way. In Section 4 we present our approach and explain how to deal with the issues mentioned before. Experimental results of a first prototypical implementation are presented in Section 5 before concluding in Section 6.

2 Foundations

We introduce some technical terms (Section 2.1), describe state of the art methods for extracting an alignment (Section 2.2), and take a closer look at one them (Section 2.3).

2.1 Nomenclature

Let $\mathcal{O}_1$ and $\mathcal{O}_2$ be ontologies that have to be matched. A correspondence is a quadruple $\langle e_1, e_2, r, c\rangle$ where e and e' are entities defined in $\mathcal{O}_1$ and $\mathcal{O}_2$. r is a semantic relation between e_1 and e_2. Within this paper the semantic relation will always be equivalence and e_1 and e_2 will always be classes or (data or object) properties. The numerical value c is referred to as confidence value. The higher the value, the higher is the probability that $r(e_1, e_2)$ holds. The confidence value is an optional element and will sometimes be omitted. The outcome of a matching system is a set of correspondences between $\mathcal{O}_1$ and $\mathcal{O}_2$. Such a set is called an alignment $\mathcal{A}$ between $\mathcal{O}_1$ and $\mathcal{O}_2$.

In the following we distinguish between linguistic entities (labels and tokens) and ontological entities (classes and properties) using the following naming convention.

`n#ClassOrProperty` - Refers to a class or property in $\mathcal{O}_n$ (with $n \in 1, 2$).
`n:Label` - Refers to a label used in $\mathcal{O}_n$ as a class or property description.
`n:Token`$_t$ - Refers to a token that appears as a part of a label in $\mathcal{O}_n$.

We will later, e.g., treat `1#AcceptedPaper` and `1:AcceptedPaper` as two different entities. The first entity appears in logical axioms and the second might be a description of the first entity. The label consists of the tokens `1:Accepted`$_t$ and `1:Paper`$_t$. We need three types of entities (logical entities, labels, tokens) because a logical entity can be described by several labels and a label can be decomposed in several tokens.

2.2 Alignment Extraction

The easiest way for selecting a final alignment $\mathcal{A}$ from a set of matching hypotheses $\mathcal{H}$ is the application of a threshold. However, a threshold does not take into account any dependencies between correspondences in $\mathcal{H}$. Thus, it might happen that an entity `1#e` is mapped on `2#e'` and `2#e''` even though `2#e'` and `2#e''` are located in different branches of the concept hierarchy.

This can be solved easily. We first sort $\mathcal{H}$ by confidence scores. Starting with an empty alignment $\mathcal{A}$, we iterate over $\mathcal{H}$ and add each $\langle e_1, e_2, =, c\rangle \in \mathcal{H}$ to $\mathcal{A}$ if $\mathcal{A}$ does not yet contain a correspondence that links one of e_1 or e_2 to some other entity. This ensures that $\mathcal{A}$ is finally a one-to-one alignment. Similar algorithms can be applied to ensure that certain anti-pattern (e.g., Asmov [5]) are avoided when adding correspondences to $\mathcal{A}$. It is also possible to use reasoning to guarantee the coherence of the

generated alignment (e.g., Logmap [6]). Checking a set of patterns is then replaced by calling a reasoning engine.

Such an approach needs to decide upon the order in which correspondences are iterated over because different orders can lead to different results. Global methods try to overcome this problem. Similarity flooding [10], for example, is based on the following assumption: The similarity between two entities linked by a correspondence in $\mathcal{H}$ must depend on the similarity of their adjacent nodes for which an initial similarity is specified in $\mathcal{H}$. The algorithm does not select a subset of $\mathcal{H}$ as final outcome but generates a refined similarity distribution over $\mathcal{H}$. Other global methods explicitly define an optimization problem in which a subset from $\mathcal{H}$ needs to be chosen that maximizes an objective function. This is detailed in the following section.

2.3 Global Optimization with Markov Logic

In [13] and [2] Markov Logic has been proposed to solve the alignment extraction problem. The authors have argued that the solution to a given matching problem can be obtained by solving the maximum a-posteriori (MAP) problem of a ground Markov logic network. In such a formalization the MAP state, which is the solution of an optimization problem, corresponds to the most probable subset $\mathcal{A}$ of $\mathcal{H}$. In the following we explain the basic idea of the approach proposed in [13]. Due to the lack of space we omit a theoretical introduction to Markov Logic and refer the reader to [15].

In [13] the authors have defined, due to the fact that Markov Logic is a log linear probabilistic model, the objective function as the confidence total of $\mathcal{A} \subseteq \mathcal{H}$. Without any further constraints and given that all confidences are positive it follows that $\mathcal{A} = \mathcal{H}$. However, some of the constraints that have been mentioned above can easily be encoded as first-order formulae in Markov Logic. We can postulate that a pair of correspondences violating the 1:1 constraint is not allowed in the final solution. This can be expressed as follows.

$$map(e_1, e_2) \wedge map(e_1', e_2') \wedge e_1 = e_1' \rightarrow e_2 = e_2'$$

Similarly, coherence constraints can be added to avoid certain patterns of incoherent mappings. An example is the constraint that the classes e_1 and e_1' where e_1' is a subclass of e_1 cannot be mapped on e_2 and e_2' where e_2 and e_2' are disjoint:

$$sub(e_1, e_1') \wedge dis(e_2, e_2') \rightarrow \neg(map(e_1, e_2) \wedge map(e_1', e_2'))$$

Due to the lack of space, we cannot specify all constraints of the complete formalization. Additional constraints are required to take into account that properties can also be involved in logical inconsistencies (see [13]). Moreover, there are some soft constraints that reward homomorphism introduced by the selected correspondences.

Given such a formalization, a reasoning engine for Markov Logic can be used to compute the MAP state which corresponds to the most probable consistent mapping. In our terminology we call this mapping a global optimal solution. Note that the entities that appear in such a formalization are logical entities (classes and properties) only, while labels or token are completely ignored. The have only been used to compute weights for the matching hypotheses, which are the weights attached to the map-atoms.

3 Illustrating Examples

In Section 3.1 and 3.2 we analyze examples that illustrate problems of the classical approaches described in the previous section. In Section 3.3 we discuss the possibility to cope with these problems without introducing a new modeling style.

3.1 Multiple Token Occurrences

For most matching problems some of the tokens used in the labels will appear in more than one label. This is in particular the case for compound labels that can be decomposed into modifier and head noun. Figure 1 shows a typical example.

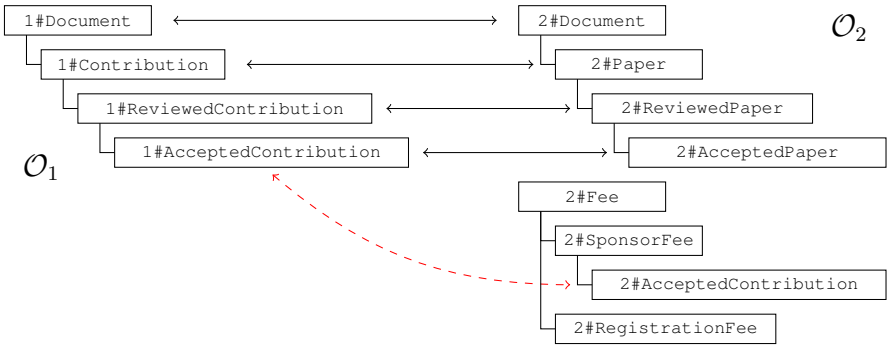


Fig. 1. Example of a non-trivial matching problem.

Let us first discuss a simplified version of the example where we ignore the branch in $\mathcal{O}_2$ rooted at the `2#Fee` class. Note that a matching problem very similar to the simplified example can be found in the OAEI conference dataset (testcase conference-ekaw). For this small excerpt there are four correspondences (solid arrows) in the reference alignment. Probably, most systems would generate $\langle$`1#Document`, `2#Document`, $=\rangle$ due to the usage of the same label. The same does not hold for the other three correspondences. For two of them the labels can be decomposed into modifier and headnoun. For all of these correspondences it is crucial to answer the question whether the words `1:Contribution` and `2:Paper` have the same meaning. How would a standard approach deal with this example? In such an approach a similarity metric would be used to compute a similarity for all relevant pairs of words. This would probably also result in a (numerical) similarity for the pair $\langle$`1:Contribution`, `2:Paper`$\rangle$, for example $sim($`1:Contribution`, `2:Paper`$) = 0.3$. This similarity would then be aggregated into a score that might result into a set of weighted hypotheses $\mathcal{H}$.

$$c_1 = \langle \texttt{1\#Document}, \texttt{2\#Document}, =, 1.0 \rangle$$
$$c_2 = \langle \texttt{1\#Contribution}, \texttt{2\#Paper}, = 0.3 \rangle$$
$$c_3 = \langle \texttt{1\#ReviewedContribution}, \texttt{2\#ReviewedPaper}, = 0.65 \rangle$$
$$c_4 = \langle \texttt{1\#AcceptedContribution}, \texttt{2\#AcceptedPaper}, = 0.65 \rangle$$

At this stage we have lost the dependency between our final decision and the question whether or not the words `1:Contribution` and `2:Paper` have the same meaning. Without being aware of this dependency it might happen that c_1, c_3, c_4 and not c_2 are selected. This would, obviously, be an inconsistent decision, because the selection of c_3 and c_4 should always result in the selection of c_2.

One might criticize that we are making (invalid) assumptions. Above we used the average for aggregating confidences. One might also use, for example, the minimum. This results in the same confidences for c_2, c_3 and c_4. Nevertheless, the distance between `1:Contribution` = `2:Paper` is taken into account not once but several times. Thus, the decision related to c_2 will not be affected by the possibility of generating c_3 and c_4, while a human expert would take c_3 and c_4 into account.

Let us now analyze the extended example where we have the additional branch that deals with fees and (monetary) contributions. Now we have another (incorrect) matching candidate.

$$c_5 = \langle \texttt{1\#AcceptedContribution}, \texttt{2\#AcceptedContribution}, =, 1.0 \rangle$$

Obviously, c_5 is in a 1:1 conflict with c_4. A consistent 1:1 mapping might thus consist of c_1, c_2, c_3 and c_4 or (exclusive!) c_5. However, taking the involved tokens and their possible meanings into account, we should not generate an alignment that contains c_2 and c_5 at the same time. Such an alignment will only be correct, if the tokens in $\mathcal{O}_1$ are used in an inconsistent way.

The classical approach cannot handle such cases in the appropriate way. As long as the tokens themselves are not explicitly modeled as entities in the extraction phase, unreasonable and inconsistent decisions, inconsistent with respect to assumptions related to the use of words, are made.

3.2 Ignoring Modifiers

We illustrate another pattern by an example taken from the OAEI conference dataset, namely the confof-ekaw testcase. The reference alignment for this testcase contains 20 correspondences, here we are interested in the following three correspondences.

$$\langle \texttt{1\#Banquet}, \texttt{2\#ConferenceBanquet}, = \rangle$$
$$\langle \texttt{1\#Participant}, \texttt{2\#ConferenceParticipant}, = \rangle$$
$$\langle \texttt{1\#Trip}, \texttt{2\#ConferenceTrip}, = \rangle$$

The developer of $\mathcal{O}_2$ was more verbose than the developer of $\mathcal{O}_1$. In $\mathcal{O}_2$ some of the labels have been extended by adding the prefix modifier `2:Conference`. This modifier has been omitted in $\mathcal{O}_1$ because each of the participants, trips and banquets is implicitly always associated to a conference. We are not interested in pros and cons of both styles. Both exist and a matching system should be able to cope with them.

Let us again think how we, as reasonable agents, would deal with this issue. After studying the $\mathcal{O}_1$ ontology, we would come to the decision, that it might make sense to ignore the token $\texttt{1:Conference}_t$ whenever it appears as modifier. Maybe we would first try to match both ontologies without ignoring the modifier, then we would

match both ontologies while ignoring $\texttt{1:Conference}_t$ when it appears as modifier. In both cases we ensure the coherency of the generated alignment. For our example the outcome would be that the second approach allows to generate three additional correspondences that do not introduce any logical conflicts. Thus, ignoring the modifier `1:Conference` seems to be a good choice.

Again, we can see that a first class citizen in such considerations are linguistic entities. We make certain decisions about the role of tokens and their implications result in the acceptance of correspondences, while logical constraints that deal with ontological entities have also an impact on our interpretation of tokens.

3.3 Work Around

In [12] the authors have proposed a measure called extended Tversky similarity that copes with the situation described in Section 3.2. Their idea is to weigh each token by its information content. A token like `2:Conference` that appears very often has a very low weight. It follows that a relatively high confidence score is assigned to a correspondence like $\langle \texttt{1\#Banquet}, \texttt{2\#ConferenceBanquet}, = \rangle$ because `2:Conference` has only a limited discriminative power. Note that this approach is still based on the principle to assign confidences to correspondences. Once this assignment has been made, the tokens that have been involved are no longer taken into account.

This technique has been implemented in the YAM++ matcher. This matcher achieved very good results the OAEI 2012 campaign [1] (see also the results table in Section 5). However, not the number of token-occurrences is important, but the maximal number of additional coherent correspondences that would result from ignoring a modifier. While these numbers are often correlated, this is not necessarily the case. Suppose that we have an ontology that contains the class `1#PaperAuthor` and the property `1#paperTitle`, as well as some other labels that contain the token $\texttt{1:paper}_t$. Let the other ontology contain a class `2#Author` (including authors of reviews) and a property `2#title` (to describe the title of a conference). In $\mathcal{O}_1$ we have a relatively high number of $\texttt{1:paper}_t$-token occurrences, however, the word $\texttt{1:paper}_t$ is in most cases a feature that needs to be taken into account. This can be derived from the fact that $\langle \texttt{1\#PaperAuthor}, \texttt{2\#Author}, = \rangle$ and $\langle \texttt{1\#paperTitle}, \texttt{2\#title}, = \rangle$ cannot be added without introducing logical conflicts given a meaningful axiomatization in $\mathcal{O}_1$ and $\mathcal{O}_2$. In our approach we will be able to take such cases into account.

4 Approach

We first present our approach and it formalization in Section 4.1 followed by an analysis of its impact in Section 4.2 where we revisit the examples of the previous section.

4.1 Formalization

In the following we distinguish explicitly between entities from two different layers. The first layer is the layer of labels and tokens; the entities that appear in the second layer are classes and properties. In our approach we treat entities from both layers as first

class citizens of an optimization problem. Thus, we can define the objective function of our optimization problem on top of token similarities (first layer) instead of using confidence values attached to correspondences (second layer).

Hidden predicates	
$map(e_1, e_2)$	e_1 is mapped on e_2, i.e. $\langle e_1, e_2, = \rangle \in \mathcal{A}$
$equiv_t(t_1, t_2)$	t_1 and t_2 have the same meaning
$equiv_l(l_1, l_2)$	l_1 and l_2 have the same meaning
$ignore(t)$	token t can be ignored if it appears as a modifier
Logical predicates	
$sub(e_1, e_2)$	class/property e_1 is subsumed by class/property e_2
$dis(e_1, e_2)$	e_1 and e_2 are disjoint classes
$dom(e_1, e_2)$	class e_1 is the domain of property e_2
$ran(e_1, e_2)$	class e_1 is the range of property e_2
Linguistic predicates	
$pos1(l, t)$	label l has token t at first position
$pos2(l, t)$	label l has token t at second position
$pos3(l, t)$	label l has token t at third position
$has1Token(l)$	label l is composed of one token
$has2Token(l)$	label l is composed of two tokens
$has3Token(l)$	label l is composed of three tokens
$hasLabel(e, l)$	entity e is described by label l

Table 1. Variables starting with e refer to classes or properties, e.g., `1#ConferenceFee`; l refers to complete labels, e.g., `1:ConferenceFee`, and t refers to tokens, e.g., $\texttt{1:Fee}_t$

We extend the approach described in Section 2.3, i.e., we use Markov Logic and most of the constraints presented above. However, we also need a rich set of (new) predicates listed in Table 1 to support our modeling style. The first four predicates in the listing are hidden predicates. This means that we do not know in advance if the ground atoms for these predicates are true or wrong. We attach a weight in the range $[-1.0, 0.0]$ to the atoms instantiating the $equiv_t$ predicate, if we have some evidence that the respective tokens have a similar meaning. We explicitly negate the atom if there is no such evidence. As a result we have a fragment as input that might look like this.

$$
\begin{aligned}
equiv_t(\texttt{1:Accepted}_t, \texttt{2:Accepted}_t)\ &, 0.0 \\
equiv_t(\texttt{1:Organization}_t, \texttt{2:Organisation}_t)\ &, -0.084 \\
equiv_t(\texttt{1:Paper}_t, \texttt{2:Contribution}_t)\ &, -0.9 \\
\neg equiv_t(\texttt{1:Accepted}_t, \texttt{2:Rejected}_t)\ \ &unweighted
\end{aligned}
$$

We do not add any (weighted or unweighted) groundings of the map, $equiv_l$, and $ignore$ predicates to the input. Our solution will finally consist of a set of atoms that are groundings of the four hidden predicates. While we are mainly interested in the map-atoms (each atom refers to a correspondence), the groundings of the other predicates can be seen as additional explanations for the finally generated alignment. These atoms

inform us which tokens and labels are assumed to be equivalent and which tokens have been ignored.

The other predicates in the table are used to describe observations relevant for the matching problem. We describe the relations between tokens and labels and the relation between labels and logical entities.

$$pos1(\texttt{1:AcceptedPaper}, \texttt{1:Accepted}_t)$$
$$pos2(\texttt{1:AcceptedPaper}, \texttt{1:Paper}_t)$$
$$has2Token(\texttt{1:AcceptedPaper})$$
$$hasLabel(\texttt{1\#AcceptedPaper, 1:AcceptedPaper})$$

We postulate that a label is matched if and only if all of its tokens are matched. We specify this explicitly for labels of different size.[1] The 2-token case is shown here.

$$has2Token(l_1) \wedge has2Token(l_2) \wedge pos1(l_1, t_{11}) \wedge pos2(l_1, t_{12}) \wedge$$
$$pos1(l_2, t_{21}) \wedge pos2(l_2, t_{22}) \rightarrow (equiv_l(l_1, l_2) \leftrightarrow equiv_t(t_{11}, t_{21}) \wedge equiv_t(t_{12}, t_{22}))$$

Next, we have to establish the connection between label and logical entity. A logical entity is matched if and only if at least one of its labels is matched.

$$map(e_1, e_2) \leftrightarrow \exists l_1 \, \exists l_2 \, (hasLabel(e_1, l_1) \wedge hasLabel(e_2, l_2) \wedge equiv_l(l_1, l_2))$$

We follow the classic approach and translate (a subset of) the ontological axioms to our formalism by using the logical predicates. We add several constraints as restrictions of the map-predicate ensuring that the generated alignment is a 1:1 mapping and that this mapping is coherent taking the ontological axioms into account. These constraints have already been explained in [13] and we can integrate them easily in our approach as constraints on the second layer. In addition to the 1:1 constraint for the map predicate, we also add a 1:1 constraint for the $equiv_t$-predicate on the token layer. This ensures that $equiv(\texttt{1:Paper}_t, \texttt{2:Contribution}_t)$ and $equiv(\texttt{1:Contribution}_t, \texttt{2:Contribution}_t)$ cannot be true at the same time.

Computing the MAP state for the modeling described so far will always yield an empty result, because the summands in the objective function are only the weights attached to the $equiv_t$-atoms. All of them are ≤ 0, thus, the best objective will be 0, which is the objective of an empty mapping. We have to add a weighted rule that rewards each correspondence, i.e., a rule that rewards each instantiation of the map predicate. We have set the reward to 0.5.

$$map(e_1, e_2), +0.5$$

Now each correspondence added to the solution increases the score of the objective by 0.5. At the same time each instantiation of the map predicate forces to instantiate at least one $equiv_l$-atom, which again forces to instantiate the related $equiv_t$-atoms weighted with values lower or equal to zero. Thus, we have defined a non trivial optimization

[1] We have not included labels with more than three tokens in our first implementation. For larger labels, we decided to match these labels directly if they are the same after normalization.

problem in which the idea of generating a comprehensive alignment conflicts with our assumptions related to the meaning of words.

Finally, we need to explain the role of the $ignore$ predicate. We want to match a 1-token label to a 2-token label if and only if we are allowed to ignore the modifier of the 2-token label and if the remaining token is equivalent to the token of the 1-token label. This can be expressed as follows.

$$has1Token(l_1) \wedge has2Token(l_2) \wedge pos1(l_1, t_{11}) \wedge pos1(l_2, t_{21}) \wedge \\ pos2(l_2, t_{22}) \rightarrow (equiv_l(l_1, l_2) \leftrightarrow equiv_t(t_{11}, t_{22}) \wedge ignore(t_{21}))$$

However, a modifier should not be ignored be default. For that reason we have to add again a simple weighted rule.

$$ignore(t), -0.95$$

Together, with the previous constraint this rule assigns a punishment to ignoring a token that is used as modifier. Note that the weight is set to a value lower than -0.5. By setting the value to -0.95 it will only pay off to ignore a token if it will result in at least two additional correspondences ($n \times 0.5 - 0.95 > 0.0$ for $n \geq 2$).

4.2 Impact

For the small fragment depicted in Figure 1 (from Section 3.1), we present the weighted input atoms (marked with an I) and the resulting output atoms (marked with an O) in the following listing. We omit the atoms describing the relations between tokens, labels, and logical entities, as well as those that model the logical axioms.

I	O	$equiv_t(\texttt{1:Document}_t, \texttt{2:Document}_t)$	*input weight 0.0*
I	O	$equiv_t(\texttt{1:Reviewed}_t, \texttt{2:Reviewed}_t)$	*input weight 0.0*
I	O	$equiv_t(\texttt{1:Accepted}_t, \texttt{2:Accepted}_t)$	*input weight 0.0*
I		$equiv_t(\texttt{1:Contribution}_t, \texttt{2:Contribution}_t)$	*input weight 0.0*
I	O	$equiv_t(\texttt{1:Contribution}_t, \texttt{2:Paper}_t)$	*input weight -0.9*
	O	$equiv_l(\texttt{1:Document}, \texttt{2:Document})$	
	O	$equiv_l(\texttt{1:Contribution}, \texttt{2:Paper})$	
	O	$equiv_l(\texttt{1:ReviewedContribution}, \texttt{2:ReviewedPaper})$	
	O	$equiv_l(\texttt{1:AcceptedContribution}, \texttt{2:AcceptedPaper})$	
	O	$c_1 \approx map(\texttt{1\#Document}, \texttt{2\#Document})$	
	O	$c_2 \approx map(\texttt{1\#Contribution}, \texttt{2\#Paper})$	
	O	$c_3 \approx map(\texttt{1\#ReviewedContribution}, \texttt{2\#ReviewedPaper})$	
	O	$c_4 \approx map(\texttt{1\#AcceptedContribution}, \texttt{2\#AcceptedPaper})$	

The generated solution consists of four $equiv_t$-atom, four $equiv_l$-atoms, and four map-atoms. The four map-atoms are converted to the four correspondences of the output alignment $\{c_1, c_2, c_3, c_4\}$. The objective of this solution is $1.1 = 4 \times 0.5 + 0.0 + 0.0 + 0.0 + 0.0 - 0.9$. The example shows that the low similarity between $\texttt{1:Paper}_t$ and

2:Contribution$_t$ atom is compensated by the possibility to generate four correspondences. The same result would not have been achieved by attaching aggregated weights directly to the map-atoms.

Let us compare this solution to other possible and impossible solutions. Thus, let $c_5 \approx map(\texttt{1\#AcceptedContribution}, \texttt{2\#AcceptedContribution})$ and let $c_6 \approx map(\texttt{1\#Contribution}, \texttt{2\#AcceptedContribution})$. .

$$\begin{aligned}
\text{objective for } \{c_1, c_2, c_3, c_4\} &= 4 \times 0.5 - 0.9 = 1.1 \\
\text{objective for } \{c_1, c_5\} &= 2 \times 0.5 = 1.0 \\
\{c_1, c_2, c_3, c_4, c_5\} &\quad \textit{is invalid against 1:1 constraint on the token layer} \\
\text{objective for } \{c_1\} \text{ or } \{c_5\} &= 1 \times 0.5 = 0.5 \\
\text{objective for } \{c_1, c_6\} &= 2 \times 0.5 - 0.95 = 0.05
\end{aligned}$$

The alignment $\{c_1, c_5\}$ is listed with a relatively high objective. Note that $\{c_1, c_5\}$ would be invalid, if we there would be a disjointness statement between 2#Fee and 2#Document due a constraint on the layer of ontological entities. We have also added $\{c_1, c_6\}$ to our listing. It illustrates the possibility to ignore a modifier. However, this solution has a low objective and there are other solutions with a better objective.

5 Preliminary Evaluation Results

In the following we report about experiments with a prototypical implementation based on the formalization presented above. The formalization is extended as follows.

- We added the constraint that if a property p is matched on a property p', then the domain (range) of p has to be matched to the domain of p' or to a direct super or subclass of the domain (range) of p'. In the latter case a small negative weight is added to the objective.
- We derived alternative labels from the directly specified labels by ignoring certain parts. For example, we added the label 1:writes to a property labeled with 1:writesPaper, if 1:Paper was the label of that properties domain.
- We derived alternative labels by adding 1:ConferenceMember as alternative label given a label like 1:MemberOfConference.
- We added rules that allow to match two-token labels on three-token labels in case that all tokens from the two-token label are matched, however, such a case was punished with a negative weight.

We use the following basic techniques for computing the input similarity scores. First we normalize and split the labels into tokens. Given two tokens t_1 and t_2, we compute the maximum of the values returned by the following five techniques. (1) We assign a score of 0.0, if $t_1 = t_2$. (2) If t_1 and t_2 appear in the same synset in WordNet [11], we assign a score of -0.01. (3) We compute the Levenshtein distance [7], multiply it with -1 and assign any score higher than -0.2 to detect spelling variants. (4) If t_1 or t_2 is a single letter token and t_1 starts with t_2 or vice versa, we assign a score of -0.3. (5) We check if t_1 and t_2 have been modified at least two times by the same modifier. If this is the case, we assign a (very low) score of -0.9.

We have used the RockIt [14] Markov Logic engine to solve the optimization problem. RockIt does not support all logical constructs of our formalization directly. Thus, we had to rewrite existential quantification in terms of a comprehensive grounded representation. We applied our approach to the OAEI conference track. The results are depicted in Table 2.

2014	Pre	F	Rec	**2013**	Pre	F	Rec	**2012**	Pre	F	Rec
*	.80	.68	.59	YAM++ [12]	.78	.71	.65	YAM++	.78	.71	.65
AML [4]	.80	.67	.58	*	.80	.68	.59	*	.80	.68	.59
LogMap [6]	.76	.63	.54	AML	.82	.64	.53	LogMap	.77	.63	.53
XMAP [3]	.82	.57	.44	LogMap	.76	.63	.54	CODI	.74	.63	.55

Table 2. The proposed approach (*) compared with the top systems of 2012, 2013, and 2014.

We have listed the top-3 participants of the OAEI 2012, 2013, and 2014 conference track. The results are presented in term of precision (Pre), recall (Rec), and F-measure (F) using the the `ra2` reference alignment.[2] For each year the results are ordered by the F-measure that has been achieved. We inserted the results of our system, marked as *, at the appropriate row. Note that the vast majority of participating systems, which perform worse, is not depicted in the table. It can be seen that our approach is on the first position in 2014 and on the second in 2013 and 2012. This is a very good result, because we spent only a limited amount of work in the computation of the ingoing similarity scores. On the contrary, we presented above a complete description in less then 10 lines. This indicates that the quality of the generated alignments is mainly based our new approach for modeling the task of selecting the final alignment from the given similarity scores.

The OAEI conference dataset can processed in less than 20 minutes on a standard laptop. While slightly larger matching tasks are still feasible, significantly larger tasks cannot be solved anymore. Scalability is indeed an open challenge for the proposed approach. Currently we are working on a robust version of our approach in order to participate in the OAEI 2015 campaign.[3]

6 Conclusion

We presented a new approach for extracting a final alignment from an initial set of matching hypotheses. We have argued by a detailed discussion of several examples that our approach makes reasonable choices in situations where classical approaches are doomed to fail. Moreover, our approach generates results in a transparent and comprehensible manner. It can, for example, be proven that any other solution with a better objective must be invalid. Moreover, the objective for any other possible solution can be

[2] The `ra2` reference alignment is not available for the public. We thank Ondřej Šváb-Zamazal, one of the track organizers, for conducting an evaluation run outside an OAEI campaign.

[3] A first implementation is available at http://web.informatik.uni-mannheim.de/mamba/

computed to understand why the generated alignment was preferred over an alternative. A preliminary evaluation has shown that our approach can compete with the top systems participating in previous OAEI campaigns even though we put only limited effort in the optimal choice and design of the similarity measures we used in our evaluation. While the evaluation revealed that scalability is a crucial issue for the proposed approach, the positive results observed so far as well as the elegant nature of the approach engages us to improve the approach and to analyze it future work.

References

1. José-Luis Aguirre, Kai Eckert, Jérôme Euzenat, Alfio Ferrara, Willem Robert van Hage, Laura Hollink, Christian Meilicke, Andriy Nikolov, Dominique Ritze, François Scharffe, Pavel Shvaiko, Ondrej Sváb-Zamazal, Cássia Trojahn dos Santos, Ernesto Jiménez-Ruiz, Bernardo Cuenca Grau, and Benjamin Zapilko. Results of the ontology alignment evaluation initiative 2012. In *Proceedings of the 7th International Workshop on Ontology Matching*, 2012.
2. Sivan Albagli, Rachel Ben-Eliyahu-Zohary, and Solomon E. Shimony. Markov network based ontology matching. *Journal of Computer and System Sciences*, 78(1):105–118, 2012.
3. Warith Eddine Djeddi and Mohamed Tarek Khadir. XMap++: Results for oaei 2014. In *Proceedings of the 9th International Workshop on Ontology Matching co-located with the 13th International Semantic Web Conference*, pages 163–169.
4. Daniel Faria, Catia Pesquita, Emanuel Santos, Matteo Palmonari, Isabel Cruz, and Francisco Couto. The agreementmakerlight ontology matching system. In *On the Move to Meaningful Internet Systems: OTM 2013 Conferences*, pages 527–541. Springer, 2013.
5. Yves R. Jean-Mary, E. Patrick Shironoshita, and Mansur R. Kabuka. Ontology matching with semantic verification. *Web Semantics: Science, Services and Agents on the World Wide Web*, 7(3):235–251, 2009.
6. Ernesto Jiménez-Ruiz and Bernardo Cuenca Grau. Logmap: Logic-based and scalable ontology matching. In *The Semantic Web–ISWC 2011*, pages 273–288. Springer, 2011.
7. Vladimir I Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710, 1966.
8. Christian Meilicke. *Alignment incoherence in ontology matching*. PhD thesis, University Mannheim, 2011.
9. Christian Meilicke, Jan Noessner, and Heiner Stuckenschmidt. Towards joint inference for complex ontology matching. In *AAAI (Late-Breaking Developments)*, 2013.
10. Sergey Melnik, Hector Garcia-Molina, and Erhard Rahm. Similarity flooding: A versatile graph matching algorithm and its application to schema matching. In *Data Engineering, 2002. Proceedings. 18th International Conference on*, pages 117–128. IEEE, 2002.
11. George A. Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.
12. DuyHoa Ngo, Zohra Bellahsene, and Konstantin Todorov. Extended tversky similarity for resolving terminological heterogeneities across ontologies. In *On the Move to Meaningful Internet Systems: OTM 2013 Conferences*, pages 711–718. Springer, 2013.
13. Mathias Niepert, Christian Meilicke, and Heiner Stuckenschmidt. A probabilistic-logical framework for ontology matching. In *AAAI*, 2010.
14. Jan Noessner, Mathias Niepert, and Heiner Stuckenschmidt. RockIt: Exploiting parallelism and symmetry for map inference in statistical relational models. 2013.
15. Matthew Richardson and Pedro Domingos. Markov logic networks. *Machine learning*, 62(1-2):107–136, 2006.

A Multilingual Ontology Matcher

Gábor Bella*, Fausto Giunchiglia†, Ahmed AbuRa'ed†, and Fiona McNeill*

*Heriot-Watt University, †University of Trento

Abstract State-of-the-art multilingual ontology matchers use machine translation to reduce the problem to the monolingual case. We investigate an alternative, self-contained solution based on *semantic matching* where labels are parsed by multilingual natural language processing and then matched using a language-independent knowledge base acting as an interlingua. As the method relies on the availability of domain vocabularies in the languages supported, matching and vocabulary enrichment become joint, mutually reinforcing tasks. In particular, we propose a vocabulary enrichment method that uses the matcher's output to detect and generate missing items semi-automatically. Vocabularies developed in this manner can then be reused for other domain-specific natural language understanding tasks.

1 Introduction

Classification hierarchies, tree-structured data schemas, taxonomies, and term bases are widely used around the world as simple, well-understood, semi-formal data and knowledge organisation tools. They often play a normative role both as a means for classification (of documents, open data, books, items of commerce, web pages, etc.) and as sources of shared vocabularies for actors cooperating in a given domain. Activities such as international trade and mobility rely on the interoperability and integration of such resources across languages. Cross-lingual[1] ontology matching attempts to provide a solution for creating and maintaining alignments for such use cases.

State-of-the-art matchers that evaluate as the best in the *Multifarm* cross-lingual matching tasks of OAEI [6], such as AML [1] or LogMap [9], use online translation services (typically from Microsoft or Google) in order to reduce the problem of language diversity to the well-researched problem of monolingual English-to-English matching. The success of these methods is dependent on the availability of the translation service that is being used as a black box. Still, with the constant improvement of such services, matchers using machine translation are able to provide usable results and are able to deal with a wide range of languages.

In this paper we investigate a different perspective on cross-lingual matching that considers the building and maintenance of multilingual vocabularies as part

[1] We use the term *cross-lingual matching* as a specific case of multilingual matching when ontologies in two different languages are being aligned.

of the alignment task. The method is based on the use of locally available multilingual lexical-semantic *vocabularies*. Such resources are in constant evolution and are often available on the web with a more or less wide coverage of different terminological domains.

We are motivated by three considerations: first, we set out to explore to what extent such a linguistically-oriented, non-statistical approach to cross-lingual matching can be used as a viable alternative to machine translation. Secondly, we wish to provide a natively multilingual matcher that is entirely under the control of its user and does not rely on a non-free external translator service. This is necessary for high-value applications, such as e-commerce or libraries, where quality has to remain fully under the user's control. Finally, besides using vocabularies as resources for matching, we show how the matcher's output itself can become a resource in the purpose of vocabulary enrichment. This positive feedback loop exploits mismatches for increased terminological coverage which, in turn, improves subsequent matching results. One example use case is integration of open data—available in multiple languages—for mobility applications where geographical concepts and names are matched with the *GeoWordNet* catalogue [2].

While there is existing work [7] on using post-processing to repair a matching through the enrichment of background knowledge, our goal is different: we attempt to collect missing *vocabulary elements* that can be stored and subsequently reapplied, whereas [7] finds unknown *relations* between labels that may not be reusable outside the context of the matching task.

We took as basis for our work the SMATCH semantic matcher tool, for two main reasons: first, it operates on the level of meanings of labels instead of surface techniques, which makes it a suitable tool for cross-lingual semantic comparisons. Secondly, SMATCH is designed for matching *lightweight ontologies*, semi-formal knowledge organisation structures typically used for purposes of classification, that we believe are the main focus of most real-world cross-lingual matching challenges. Lightweight ontologies, as defined in [3], are characterised by (1) having a tree structure, (2) having nodes expressed as well-formed natural language labels, (3) they assume classification semantics (the extension of a node *Italy* under a node *Literature* are documents on Italian literature), and (4) the meaning of edges is not formally defined (they may stand for *is-a*, *part-of*, etc.).

The result of this work is NuSMATCH (NuSM for short), a first step in the direction of a new-generation multilingual matcher that has built-in capabilities for cross-lingual matching and that can also be used as a multilingual vocabulary enrichment tool.

The rest of the paper is organised as follows. Section 2 presents the *multilingual knowledge base*, the core resource for our matcher. Section 3 provides a brief reminder on semantic matching and on NuSM, while section 4 details our multilingual extensions. Section 5 presents vocabulary enrichment using erroneous mappings output by the matcher. Section 6 provides evaluation results and discussion, while section 7 presents issues not yet resolved.

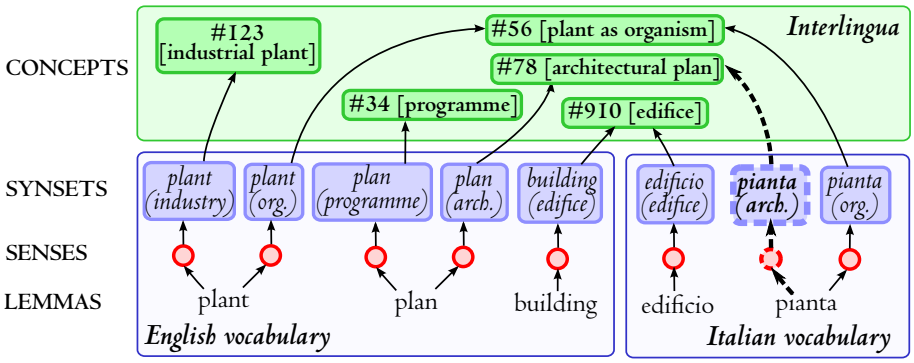


Figure 1. English and Italian vocabularies with the interlingua acting as a language-independent interoperability layer. The vocabularies may not be complete: the Italian sense and synset *pianta*, meaning 'architectural plan', is marked with dashed lines to indicate that it is missing from the Italian vocabulary.

2 A Multilingual Knowledge Base as Interlingua

Our approach to cross-lingual matching relies on a multilingual knowledge resource consisting of two layers: (1) a lower layer of multilingual *vocabularies* that are WordNet-like lexical-semantic resources; and (2) the *interlingua*: a language-independent ontology of concepts, each one linked to its corresponding vocabulary items in each language. This architecture has already been implemented at the University of Trento as part of a larger knowledge resource called the *Universal Knowledge Core* (UKC) [3], that we reuse for our purposes.

The architecture of a *vocabulary* is similar to that of Princeton WordNet [10], consisting of *lemmas* (i.e., dictionary forms of words of a language) associated to formally defined *word senses*. Synonymous senses are grouped together in synonym sets or *synsets*. Both senses and synsets are interconnected by lexical-semantic relations. Synsets represent an abstraction from the language-specific lexicon towards units of meaning and, indeed, the WordNet synset graph is sometimes used as an upper ontology for general reasoning tasks. This practice is suboptimal because of the known Anglo-Saxon cultural and linguistic bias of the synset graph (see, for example, [12]). As a solution, our multilingual knowledge base (simply *knowledge base* in the following) introduces the *interlingua* as a manually curated ontology representing a language-independent abstraction from the synset graph. Each synset in each vocabulary is mapped to a concept (fig. 1). The opposite is not necessarily true, e.g., when a vocabulary is incomplete. The interlingua acts as an interoperability layer across language-specific vocabularies, a feature that we use for cross-lingual matching.

High-quality vocabularies are costly to build in terms of human effort. Existing wordnets[2]—that we reuse to bootstrap our vocabularies when it is legally and technically possible—tend to be incomplete to a smaller or greater extent: for

[2] `http://globalwordnet.org/wordnets-in-the-world/`

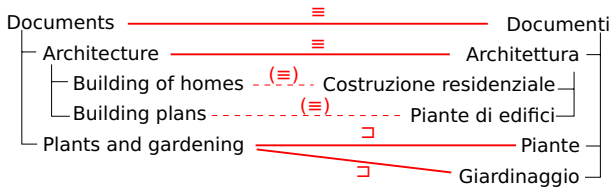


Figure 2. Example English and Italian classifications of documents, with some example mapping relations. Dashed lines with '(≡)' denote false negatives (mappings not found by the matcher), for reasons explained in section 5.

example, the Spanish *Multilingual Central Repository 3.0*[3] contains 56K lemmas and 38K synsets, the *Italian MultiWordNet*[4] contains 42K lemmas and 33K synsets, while Princeton WordNet 3.0 contains about 200K and 118K, respectively. Furthermore, wordnets tend to be general-purpose vocabularies that lack domain-specific terminology.

Efforts parallel to ours for building multilingual knowledge resources do exist. In earlier efforts such as EuroWordNet [11] or MCR [4] cross-lingual interoperability was provided by mapping non-English synsets to their English Princeton WordNet counterparts. This meant inheriting the English-centric lexical-semantic bias both in vocabulary construction and in reasoning. *BabelNet* [5] is a more recent and more advanced effort, with the same architectural design and underlying ideas as our knowledge base. The difference lies in the methodology of building it: BabelNet is mostly built automatically from diverse sources such as *Wikipedia* and *OmegaWiki*, while our knowledge base is built and maintained by human effort using both expert input and crowdsourcing. While the general problem of constructing lexical-semantic resources is beyond the scope of this paper, one of the outcomes of our work is a method for vocabulary enrichment using the output of NuSM.

3 NuSM

NuSM is designed as a multilingual extension of the SMATCH (English-only) semantic matcher [8]. Matching is semantic because, first, it is based on word senses extracted from ontology labels, secondly, it is performed using propositional logical inference and, thirdly, the mappings returned are description logic relations of equivalence, subsumption, and disjointness (for an example see fig. 2). We follow the basic four-step design of SMATCH, shown as pseudocode in fig. 3. Two new pre- and post-processing steps were added for language detection and for the semi-automated enrichment of vocabularies, respectively.

Below we provide a brief overview of each step of the matching process, followed by an in-depth discussion on the steps that are new or were modified.

[3] `http://adimen.si.ehu.es/web/MCR`
[4] `http://multiwordnet.fbk.eu`

	SMATCH	NuSM
step 0		`srcLang := detectLanguage(srcTree)` `trgLang := detectLanguage(trgTree)`
step 1	`computeLabelFormulas(srcTree)` `computeLabelFormulas(trgTree)`	`computeLabelFormulas(srcLang, srcTree)` `computeLabelFormulas(trgLang, trgTree)`
step 2	`computeNodeFormulas(srcTree)` `computeNodeFormulas(trgTree)`	

step 3	`for each srcAtom in srcTree:`
	`  for each trgAtom in trgTree:`
	`  wordNetMatcher(srcAtom, trgAtom)`
	`  stringMatcher(srcAtom, trgAtom)`
		`for each srcAtom in srcTree:`
		`  for each trgAtom in trgTree:`
		`  conceptMatcher(srcAtom, trgAtom)`
		`  nameMatcher(srcAtom, trgAtom)`

step 4	`mappings := treeMatcher(srcTree, trgTree)`	
step 5		`enrichVocabularies(mappings)`

Figure 3. Comparison of the high-level steps in SMATCH and NuSM.

For a more detailed presentation of semantic matching and the original SMATCH tool, we refer the reader to [8].

Step 0 is a new pre-processing step that detects the language of the two trees in input. We do not handle the rare case of ontologies mixing labels in multiple languages, as this would reduce the overall accuracy of language detection. Processing is interrupted if for the detected language no suitable vocabulary or NLP parser is available.

Step 1 computes *label formulas* for the two trees, that is, a propositional description logic formula corresponding to the semantic representation of the label. Atoms of the formula are sets of concepts from the interlingua, possibly representing the meaning of the atom, while operators are conjunctions, subjunctions, and negations. For example, in fig. 2, for the English label *Plants and gardening* the formula $plant \sqcup gardening$ is computed where *plant* and *gardening* are sets of concepts and the coordinating conjunction *and* becomes a disjunction (since the node classifies documents about any of the two topics). As for the label *Building plans*, it becomes a conjunctive formula: $building \sqcap plan$. The difference with respect to SMATCH is that label formulas are computed in a language-dependent manner, while meanings associated to the atoms are language-independent concepts from the interlingua instead of WordNet synsets.

Step 2 computes for each node tree their *node formulas*, which are formulas describing labels in the context of their ancestors. This step consists of computing for each label formula its conjunction with the label formulas of all of its ancestors. For *Plants and gardening*, this becomes $(plant \sqcup gardening) \sqcap document$. This step was not modified with respect to the original SMATCH.

Step 3 collects axioms relevant to the matching task. For each meaning in each atom of the source tree, step 3 retrieves all relations that hold between it and all meanings of all atoms in the target tree. In SMATCH, WordNet is used as a knowledge base (`wordNetMatcher` method) and additional axioms are inferred through string matching techniques (`stringMatcher` method). In NuSM, the interlingua is used as background knowledge (`conceptMatcher`) and string

matching is used mainly for names (`nameMatcher`). For example, for the pair of atoms (*plant*, *pianta*) retrieved from the interlingua in fig. 1, if both have a concept set of two concepts, this means retrieving potential relations for four concept pairs.

Step 4 performs the matching task (`treeMatcher` method) by running a SAT solver on pairs of source-target node formulas (f_S, f_T), computed in step 2 and complemented by corresponding axioms retrieved in step 3. If a pair turns out to be related by one of three relations: *equivalence* $f_S \leftrightarrow f_T$, *implication* $f_S \leftarrow f_T$ or $f_S \rightarrow f_T$, or *negated conjunction* $\neg(f_S \wedge f_T)$ then the mapping relation equivalence, subsumption, or disjointness is returned as a result, respectively. If none of the above holds, a no-match (*overlap*) relation is returned. This step was not modified with respect to the original SMATCH.

Step 5 is introduced specifically for NuSM as a post-processing step. Its goal is to discover mismatches resulting from missing vocabulary items, and help extend the vocabulary accordingly. For example, in fig. 2, no relation is returned between *Building plans* and *Piante di edifici* if the meaning 'plan' for *pianta* is missing from the Italian vocabulary.

4 Cross-Lingual Matching

In this section we explain how steps 1 and 3 were extended to adapt to cross-lingual operation.

4.1 Computing Label Formulas

The `computeLabelFormulas` method consists of three substeps: (1) building the label formula by parsing each label using language-specific NLP techniques; (2) computing of concept sets for each atom of the label formula; and (3) context-based sense filtering for polysemy reduction.

In NuSM, word senses in label formulas are represented by language-independent concepts from the interlingua. In order to compute label formulas and the concept sets of its atoms, language-dependent parsing is performed on labels.

Substep 1.1: label formulas are built by recognising words and expressions that are to be represented as atoms, and by parsing the syntactic structure of the label. For this purpose we use NLP techniques adapted to the specific task of ontology label parsing, distinguished by the shortness of text (typically 1-10 words) and a syntax that is at the same time limited (mostly noun, adjective, and prepositional phrases) and non-standard (varying uses of punctuation and word order). Depending on the language, different NLP techniques are used:

- word boundaries are identified through language-dependent tokenisation, e.g., *dell'/acqua* in Italian vs. *water/'s* in English, the apostrophe falling on different sides;
- language-dependent part-of-speech tagging helps in distinguishing open- and closed-class words where the former (nouns, verbs, adjectives, adverbs) become atoms while the latter (coordinating conjunctions, prepositions, punctuation, etc.) become logical operators;

English	Italian	Operator
except, non, without, ...	eccetto, escluso, non, senza, ...	$\neg$
and, or, ',', ...	e, o, ',', ...	$\sqcup$
of, to, from, against, for, ...	di, del, della, dello, dell', a, al, alla, allo, all', per, contro, ...	$\sqcap$

Figure 4. Mapping of closed-class words in labels to description logic operators (the list is incomplete).

- lemmatisation (morphological analysis of word forms in order to obtain the corresponding lemmas) is also performed using language-dependent methods, e.g., rule-based, dictionary-based, or the combination of the two;
- multiwords (e.g., *hot dog*) are recognised using dictionary lookup in the appropriate knowledge base vocabulary;
- closed-class words (pronouns, prepositions, conjunctions, etc.) and certain punctuation are mapped to the logical operators of conjunction, disjunction, and negation where mappings are defined for each language (cf. fig. 4);
- syntactic parsing—that determines how logical formulas are bracketed—is also done in a language-dependent manner.

Substep 1.2: concept sets are computed for each atom by retrieving from the interlingua all possible language-independent concepts for each open-class word appearing in the label. Thus, for the word *plant* we retrieve both the concept *plant as organism* and the concept *industrial plant* (fig. 1). What is new with respect to SMATCH is the language-independence of concepts and that concepts of derivationally related words are also retrieved, e.g., *plantation*, *planting*. This provides us increased robustness with respect to approximate grammatical correspondences between labels, a phenomenon that we observed as much more common in the cross-lingual than in the monolingual case (e.g., *piante di banane* vs. *banana plantation*).

Substep 1.3: sense filtering. In SMATCH, two atoms are by default considered equal if they have the same word form or lemma, regardless of the actual meanings: if the word *plant* appears both in the source and the target tree, they may be matched regardless of their respective meanings (*living organism* or *industrial building*). In order to reduce false positives due to such cases of polysemy, SMATCH implements a form of word sense disambiguation called *sense filtering*. This operation has a lesser importance in a cross-lingual scenario as the coincidence of homographs across languages is much rarer. For example, matching the English word *plant* with the Italian word *pianta*, both polysemous as shown in fig. 1, does not pose a problem as *pianta* does not have a meaning of 'industrial plant', nor does *plant* mean 'architectural plan'. This phenomenon acts as a 'natural' word sense disambiguation technique, allowing us to finetune recall by switching off the sense filtering algorithm implemented in SMATCH when the source and target languages are different and only apply it if the two languages are the same.

4.2 Retrieval of Axioms

SMATCH performs semantic matching between atoms by retrieving axioms as WordNet relations between senses and synsets (the `wordNetMatcher` method in fig. 3). NuSM, in contrast, relies on language-independent ontological relations existing in the interlingua (`conceptMatcher`). Equivalence is implied by concept equality and subsumption is derived from *is-a*, *attribute-value*, and *part-whole* relations, taking transitivity into account.

String similarity is a common metric used in monolingual matchers. SMATCH relies on string similarity between words and between glosses of WordNet synsets (the `stringMatcher` method includes both techniques) whenever WordNet does not provide any semantic axioms. Even though string similarity has a more limited scope of use in cross-lingual matching—words unrecognised because missing from the vocabularies cannot be assumed to match across different languages—we still use it for the matching of names and acronyms which tend to have a higher resemblance across languages (`nameMatcher`). We discarded gloss-based matching as these are not available for all vocabularies and the gloss-based matcher does not work on glosses written in different languages.

5 Vocabulary Enrichment

Term lists, taxonomies, and classifications, when available in multiple languages, are useful resources for the extraction of domain-specific terminology. The idea is to exploit incorrect mappings in order to identify the vocabulary elements missing for a given language and, consequently, to enrich them in a semi-automated manner, supervised by a human user.

Generally, we consider that mappings perceived by the user as incorrect can be explained by thee main phenomena: (1) the incompleteness of the knowledge base, (2) the design and limitations of the matcher (e.g., NLP errors or the inability to match rough translations such as *Building of homes* vs. *Costruzione residenziale*, 'residential construction') , and (3) modelling errors in the classifications themselves (example: *Gardening and landscaping* classified under *Gardening* results in two being inferred to be equivalent due to classification semantics).

In the following we concentrate on errors of type 1 and especially on missing vocabulary items: word forms, lemmas, senses, and synsets. We leave the problem of enrichment of the interlingua by concepts and relations for future work. We provide a semi-automated method that identifies errors stemming from an incomplete vocabulary and proposes a corresponding repair-by-enrichment action to the user. The semi-automated approach strikes a balance between reducing human effort and maintaining the high quality of vocabularies. It requires the contribution of a skilled person, ideally a data scientist, with a good knowledge of both languages.

Step 1: selection of the tree to process. In order to detect whether vocabulary enrichment is necessary, we either rely on a decision by the user or on a heuristic based on the number of unrecognised words found in one of the trees

being over a certain threshold. The goal is to select the tree that corresponds to the vocabulary poorer in terminological coverage: in the following we will call this tree the 'poor tree' and the other one the 'rich tree'. The repair process traverses the poor tree in depth-first order from the root, as the repair of a node affects all of its descendants.

Step 2: node-by-node identification of false negative mappings. False negatives, by definition, are true mappings not found by the matcher. Our repair method, however, relies on this information to identify missing vocabulary items. For this reason, we need to have access to ground truth in the form of equivalences and subsumptions. We propose three possible methods for obtaining ground truth:

- *user-provided*, e.g., by manually pointing out false negatives node by node during the traversal process.
- *Pre-existing:* a great number of lightweight ontologies are available on the web in multiple languages, often as industry standards of economic areas englobing multiple countries (in section 6 we provide concrete examples). These multilingual classifications can be seen as *fully aligned parallel corpora* and be used for vocabulary enrichment where the alignment provides ground truth.
- *Automatically obtained:* the (monolingual) SMATCH is run in parallel using a machine translation service as preprocessor. We automate the identification of false negatives by comparing the mappings output by both SMATCH and NuSM. Negatives output by NuSM that are positives for SMATCH are likely candidates for false negatives. We assume that precision is high (false positives are few) in the monolingual case—which is generally true, cf. the evaluations in [8]—and that the overlap of the positives of SMATCH and NuSM is not total, in other words, that the former is able to provide new positives to the latter. Our experiments showed this to be the case (cf. section 6).

Step 3: identification of the missing vocabulary item and repair. As an example for the repair process, let us take the labels *Building plans* and *Piante di edifici* from fig. 2. They are represented here as atoms containing their meanings retrieved from the interlingua in fig. 1:

Building #910 'edifice'	⊓	*plans* #78 'architectural plan' #34 'programme'	≡	*Piante* #56 'plant as organism'	⊓	*edifici* #910 'edifice'

Because of the missing sense and synset 'architectural plan' for the lemma *pianta*, indicated by dashed lines in fig. 1, the equivalence is missed by the matcher. In the repair scenario, however, we are supposing it to be provided as ground truth. Once such an erroneous mapping has been identified, repair proceeds through the substeps below.

Substep 3.1: pre-selection of atoms that are likely subjects for repair. For each false negative mapping identified while traversing the poor tree, the atoms of the corresponding label are analysed. Atoms of unrecognised words (word forms or lemmas) are given priority, as an unrecognised word is a trivial cause

of false negatives. In the absence of unrecognised words, all atoms of the label are selected. In our example, the word *piante* is a recognised word (it does have one meaning, 'plant as organism', in the vocabulary), thus both $atom_{piante}$ and $atom_{edifici}$ are pre-selected.

Substep 3.2: selection of repair candidates. A repair candidate is a pair (*preselected atom*, *repair concept*) that, when the repair concept is substituted into the atom, repairs the mapping so that the mapping relation corresponds to the ground truth. In our example, ($atom_{piante}$, 'architectural plan') is such a repair candidate. In substep 2 a small subset of *repair concepts* is selected, depending on the ground truth relation to be obtained. If the relation is equivalence then the set of repair concepts corresponds to the concepts appearing in the 'rich' node formula of the mapping. If the relation is more general (resp. less general) then it corresponds to the concepts appearing in the 'rich' node formula plus all of their ancestors (resp. descendants). The suitable (*atom*, *repair concept*) pairs are retained as *repair candidates*. For the node *Piante di edifici* two repair candidates are found: ($atom_{piante}$, 'programme') and ($atom_{piante}$, 'architectural plan'). No other substitution of any concept from the left-hand side into any atom on the right-hand side leads to equivalence.

Substep 3.3: identification of the missing vocabulary item and its creation. The user filters appropriate repair candidates by answering questions such as *'is meaning "architectural plan" suitable for word* piante *in this label?'*. Upon an affirmative answer, we find the missing vocabulary item(s) within the path between the repair concept and the surface word form of the atom. Repair ends by inserting newly created item(s) into the vocabulary (again upon user acceptance). In our case, the presence of an Italian synset connected to the concept of 'architectural plan' is verified. As it is missing, a new synset is created, together with a sense and links connecting the synset with the lemma *pianta*. The created items are the ones shown in dashed lines in fig. 1.

6 Evaluation and Discussion

Our evaluations were performed on two language pairs: English-Spanish and English-Italian. We used a diverse set of industrial and public multilingual classifications and term bases.[5] As these classifications are fully aligned across languages, they provide ground truth for equivalent mappings. However, because of the nature of semantic matching, other valid equivalences and subsumptions may be returned between non-aligned nodes. For example, *Forestry/Logging* and *Forestry/Logging/Logging* are equivalent nodes according to classification semantics (both are formalised as $forestry \sqcap logging$), yet such relations are missing from our ground truth. Manual production of ground truth being beyond our means for the 2,600 nodes evaluted, we have simplified our evaluations in order to allow the automation of tests:

[5] NACE: Statistical Classification of Economic Activities in the European Community, Rev. 2 (`ec.europa.eu/eurostat/ramon/`), EUROVOC: the EU's multilingual thesaurus (`eurovoc.europa.eu`), UDC: Universal Decimal Classification (`udcc.org`).

Corpus	Lang.	# nodes per tree	Avg. label length	Avg. depth	NuSM Prec. ≡	NuSM Recall ≡	Google smatch Prec.	Google smatch Recall
EUROVOC	EN-ES	300	2.3	1	95.9%	47.0%	98.2%	73.5%
EUROVOC	EN-IT	300	2.2	1	97.7%	56.4%	97.9%	77.9%
NACE	EN-ES	880	5.9	3.5	75.9%	20.7%	82.0%	28.5%
NACE-ATECO	EN-IT	880	6.2	3.5	82.4%	20.1%	90.3%	21.7%
UDC	EN-ES	125	5.3	2.5	63.3%	24.8%	100%	19.2%
UDC	EN-IT	125	5.1	2.5	100%	20.8%	71.7%	26.4%

Figure 5. Cross-lingual evaluation results on parallel classifications. Also included are the scores obtained by the monolingual SMATCH coupled with Google Translate.

- only relations of equivalence, that is, only perfect matches are evaluated as positives (subsumptions and disjointness are discarded);
- all returned equivalences that are not in the ground truth and cannot be trivially mapped to it (by reordering labels or removing duplicate labels) are considered as false positives.

Our results are in fig. 5. We consider the scores as promising first results, especially given our conservative evaluation method. According to close scrutiny, mapping errors (false positives and negatives) were a consequence of the following factors:

- the Spanish and Italian vocabularies we used contain 32K and 42K words, respectively, unlike our 130K English vocabulary. Missing words, senses, and synsets reduce both recall and precision.
- a weak point of our current matcher is its multilingual syntactic parser, which often results in wrong bracketing in label formulas. The longer the labels the higher the probability of a parsing error, which explains the gradual performance degradation correlated with increased label lengths in our evaluation datasets.
- the most important cause of low recall figures is the high number of non-exact translations present in the data (similar to the example *Building of homes* vs. *Costruzione residenziale*) in fig. 2). Such linguistic 'fuzziness' is perhaps the hardest cross-lingual matching problem to tackle.

The last two columns in fig. 5 represent scores obtained by SMATCH when fed by Google-translated English text. These scores are somewhat higher, although by varying margins and not in all cases. This is explained by radically different underlying NLP techniques: machine translators are essentially statistical tools based on word n-grams and thus work well on rough translations where no word-by-word cross-lingual correspondence exists. On the other hand, the statistical nature of machine translation sometimes introduces translation errors. The hypothesis that the two different approaches yield partly different matching results is confirmed by preliminary quantitative evaluations that gave 38.7% (EUROVOC), 55.3% (NACE), and 45.8% (UDC) as the percentage of true positives that were *not* found by NuSM among those that *were* found by Google-SMATCH. This proves that the translation-based method for obtaining ground truth that we supposed in section 5 can effectively work.

7 Conclusions and Future Work

The results presented in this paper, both regarding cross-lingual matching and vocabulary enrichment, reflect work in progress, with improvements ongoing in several areas. Improved language-specific syntactic parsing of ontology labels is likely to have a big impact on our scores. In the repair method, we plan to extend the scope of repair to the interlingua, both to concepts and relations. Finally, given our results, we see a new line of research in combining the vocabulary-based technique presented here with machine translation. Our observation on the difference between the sets of true positives returned by the two techniques points in the direction of a potentially efficient ensemble method.

Acknowledgment We owe a big thanks to Aliaksandr Autayeu, one of the main developers and the current maintainer of monolingual SMATCH, for his advice and for his relentless work on keeping the tool up to date. We also acknowledge the *SmartSociety* project, funded by the 7th Framework Programme of the European Community.

References

1. Daniel Faria et al. The AgreementMakerLight Ontology Matching System. In Robert Meersman et al., editor, *On the Move to Meaningful Internet Systems: OTM 2013 Conferences*, volume 8185 of *Lecture Notes in Computer Science*, pages 527–541. Springer Berlin Heidelberg, 2013.
2. Fausto Giunchiglia et al. GeoWordNet: A Resource for Geo-spatial Applications. In *Proceedings of ESWC 2010*, pages 121–136.
3. Fausto Giunchiglia et al. Faceted Lightweight Ontologies. In *Conceptual Modeling: Foundations and Applications*, volume 5600. Springer Berlin Heidelberg, 2009.
4. J. Atserias et al. The MEANING Multilingual Central Repository. In *In Proceedings of the Second International WordNet Conference*, pages 80–210, 2004.
5. Maud Ehrmann et al. Representing Multilingual Data as Linked Data: the Case of BabelNet 2.0. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014), Reykjavik, Iceland, May 26-31, 2014*.
6. Zlatan Dragisic et al. Results of the Ontology Alignment Evaluation Initiative 2014. In *ISWC 2014, Riva del Garda, Trentino, Italy.*, pages 61–104, 2014.
7. Fausto Giunchiglia, Pavel Shvaiko, and Mikalai Yatskevich. Discovering Missing Background Knowledge in Ontology Matching. In *Proceedings of ECAI 2006, Riva Del Garda, Italy.*
8. Fausto Giunchiglia, Mikalai Yatskevich, and Pavel Shvaiko. Semantic Matching: Algorithms and Implementation. *J. Data Semantics*, 9:1–38, 2007.
9. Ernesto Jiménez-Ruiz and Bernardo Cuenca Grau. LogMap: Logic-Based and Scalable Ontology Matching. In *The Semantic Web – ISWC 2011*, volume 7031, pages 273–288. 2011.
10. George A. Miller. WordNet: A Lexical Database for English. *Commun. ACM*, 38(11):39–41, November 1995.
11. Piek Vossen, editor. *EuroWordNet: A Multilingual Database with Lexical Semantic Networks*. Kluwer Academic Publishers, Norwell, MA, USA, 1998.
12. Piek Vossen, Wim Peters, and Julio Gonzalo. Towards a Universal Index of Meaning. In *SIGLEX99: Standardizing Lexical Resources*, pages 81–90, 1999.

Understanding a Large Corpus of Web Tables Through Matching with Knowledge Bases – An Empirical Study

Oktie Hassanzadeh, Michael J. Ward, Mariano Rodriguez-Muro, and Kavitha Srinivas

IBM T.J. Watson Research Center
Yorktown Heights, NY, USA
{hassanzadeh,MichaelJWard,mrodrig,ksrinivs}@us.ibm.com

Abstract. Extracting and analyzing the vast amount of structured tabular data available on the Web is a challenging task and has received a significant attention in the past few years. In this paper, we present the results of our analysis of the contents of a large corpus of over 90 million Web Tables through matching table contents with instances from a public cross-domain ontology such as DBpedia. The goal of this study is twofold. First, we examine how a large-scale matching of all table contents with a knowledge base can help us gain a better understanding of the corpus beyond what we gain from simple statistical measures such as distribution of table sizes and values. Second, we show how the results of our analysis are affected by the choice of the ontology and knowledge base. The ontologies studied include DBpedia Ontology, Schema.org, YAGO, Wikidata, and Freebase. Our results can provide a guideline for practitioners relying on these knowledge bases for data analysis.

Keywords: Web Tables, Annotation, Instance-Based Matching

1 Introduction

The World Wide Web contains a large amount of structured data embedded in HTML pages. A study by Cafarella et al. [6] over Google's index of English documents found an estimated 154 million high-quality relational tables. Subsequent studies show the value of web tables in various applications, ranging from table search [15] and enhancing Web search [1, 3] to data discovery in spreadsheet software [2, 3] to mining table contents to enhance open-domain information extraction [7]. A major challenge in applications relying on Web Tables is lack of metadata along with missing or ambiguous column headers. Therefore, a content-based analysis needs to be performed to understand the contents of the tables and their relevance in a particular application.

Recently, a large corpus of web tables has been made publicly available as a part of the Web Data Commons project [12]. As a part of the project documentation [13, 14], detailed statistics about the corpus is provided, such as distribution

of the number of columns and rows, headers, label values, and data types. In this paper, our goal is to perform a semantic analysis of the contents of the tables, to find similarly detailed statistics about the kind of entity types found in this corpus. We follow previous work on recovering semantics of web tables [15] and column concept determination [8] and perform our analysis through matching table contents with instances of large cross-domain knowledge bases.

Shortly after we started our study, it became apparent that the results of our analysis do not only reflect the contents of tables, but also the contents and ontology structure of the knowledge base used. For example, using our approach in tagging columns with entity types (RDF classes) in knowledge bases (details in Section 2), we observe a very different distribution of tags in the output based on the knowledge base used. Figure 1 shows a "word cloud" visualization of the most frequent entity types using four different ontologies. Using only DBpedia ontology classes, the most dominant types of entities seem to be related to people, places, and organizations. Using only YAGO classes, the most frequent types are similar to those from DBpedia ontology results, but with more detailed breakdown and additional types such as "Event" and "Organism" that do not appear in DBpedia results. Freebase results on the other hand are very different, and clearly show a large number of music and media related contents in Web tables. The figure looks completely different for Wikidata results, showing "chemical_compound" as a very frequent type, which is not observed in Freebase or YAGO types. This shows the important role the choice of knowledge base and ontology plays in semantic data analysis.

In the following section, we briefly describe the matching framework used for the results of our analysis. We then revise some of the basic statistics provided by authors of the source data documentation [14], and then provide a detailed analysis of the entity types found in the corpus using our matching framework. We end the paper with a discussion on the results and a few interesting directions for future work.

2 Matching Framework

In this section, we briefly describe the framework used for matching table contents with instances in public cross-domain knowledge bases. Although implementation of this framework required a significant amount of engineering work to make it scale, the methods used at the core of the framework are not new and have been explored in the past. In particular, our MapReduce-based overlap analysis is similar to the work of Deng et al. [8], and based on an extension of our previous work on large-scale instance-based matching of ontologies [9]. Here, we only provide the big picture to help understanding the results of our analysis described in the following sections.

Figure 2 shows the overall matching framework. As input, we have the whole corpus of Web Tables as structured CSV files on one hand and a set of RDF knowledge bases which we refer to as *reference knowledge* on the other hand. Based on our previous work on data virtualization [10], we turn both

(a) DBpedia Ontology Tags

(b) DBpedia YAGO Classes Tags

(c) Freebase Type Tags

(d) Wikidata Type Tags

Fig. 1. Word Cloud of Most Frequent Column Tags

the tabular data and RDF reference knowledge into a common format and store them as key-values on HDFS. For tabular data, the key is a unique URI identifying a column in an input table, and the values are the values that appear in the column. For reference knowledge input, the key is the RDF class URI, and the values are the labels of instances of that class. For example, URI `rep://webtables/23793831_0_4377639018067805567.csv/company+name` represents column with header `company+name` in file `23793831_0_4377639018067805567.csv` in the input data. The values associated with this URI are contents of the column, which in this case is a list of company names. An example of reference knowledge URI is `http://dbpedia.org/ontology/Company` which is the DBpedia ontology class representing entities of type "Company". The values associated with this URI are labels of instances of this type, which means a list of all company names in DBpedia.

The similarity analysis component of the framework takes in the key-values and returns as output a table with each record associating a column in an input table with a tag which is an RDF class in reference knowledge, along with a confidence score. This tag indicates a similarity between values associated with the column and the class in input key-values, based on a similarity measure. Our system includes a large number of similarity functions but for the purpose of this study, we focus on one similarity measure that is very simple yet accurate and powerful for annotation of tables. Similar to Deng et al. [8], we refer to this

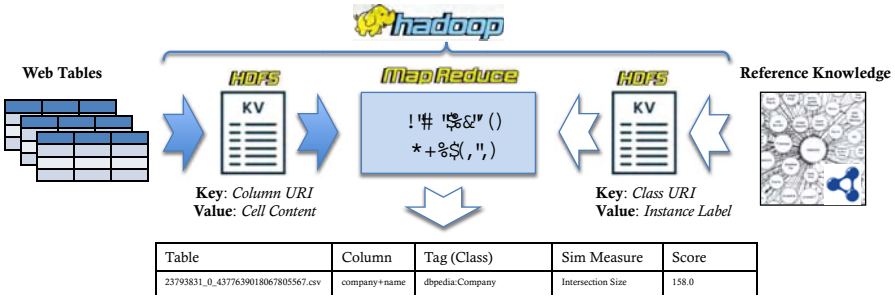


Fig. 2. Matching Framework

similarity analysis as *overlap analysis*. The values are first *normalized*, i.e., values are changed to lowercase and special characters are removed. We also filter numeric and date values to focus only on string-valued contents that are useful for semantic annotation. The similarity score is then the size of the intersection of the sets of filtered normalized values associated with the input URIs. The goal of overlap analysis is to find the number of values in a given column that represent a given entity type (class) in the input reference knowledge. In the above example, the column is tagged with class `http://dbpedia.org/ontology/Company` with score `158`, which indicates there are 158 values in the column that (after normalization) appear as labels of entities of type Company on DBpedia.

The *reference knowledge* in this study consists of three knowledge bases: (i) DBpedia [4] (ii) Freebase [5], and (iii) Wikidata [11, 16]. We have downloaded the latest versions of these sources (as of April 2015) as RDF NTriples dumps. DBpedia uses several vocabularies of entity types including DBpedia Ontology, Schema.org, and YAGO. We report the results of our analysis separately for these three type systems, which results in 5 different results for each analysis. We only process the English portion of the knowledge bases and drop non-English labels.

3 Basic Statistics

We first report some basic statistics from the Web Tables corpus we analyzed. Note that for this study, our input is the English subset of the Web Tables corpus [14] the same way we only keep the English portion of the reference knowledge. Some of the statistics we report can be found on the data publisher's documentation [14] as well, but there is a small difference between the numbers that could be due to different mechanisms used for processing the data. For example, we had to drop a number of files due to parsing errors or decompression failures, but that could be a results of the difference between the libraries used.

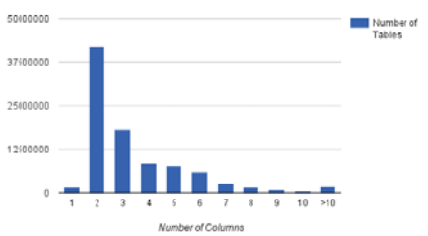
The number of tables we successfully processed is 91,357,232, that results in overall 320,327,999 columns (on average 3.5 columns per table). This results in 320,327,999 unique keys and 3,194,624,478 values (roughly 10 values per column) in the key-value input of Web Tables after filtering numerical and non-string

values for similarity analysis. DBpedia contains 369,153 classes, out of which 445 are from DBpedia Ontology, 43 are from Schema.org, and 368,447 are from YAGO. Freebase contains 15,576 classes, while Wikidata contains 10,250 classes. The number of values after filtering numeric and non-string values is 67,390,185 in DBpedia, 169,783,412 in Freebase, and Wikidata has 2,349,915 values. These numbers already show how different the knowledge bases are in terms of types and values.

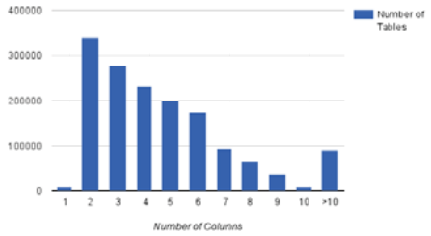
We first examine the distribution of rows and columns. Figure 3(a) shows the overall distribution of columns in the Web Tables. As it can be seen, the majority of the tables have lower than 3 columns. There are 1,574,872 tables with only 1 column, and roughly 62 million out of the 91 million tables (32%) have 2 or 3 columns. Now let us consider only the tables that appear in the output of our overlap analysis with intersection threshold set to 20, i.e., tables that in at least one of their columns have more than 20 normalized values shared with one of the knowledge reference sources. Such tables are much more likely to be of a higher quality and useful for further analysis and applications. Figure 3(b) shows the distribution of columns over these tables. As the figure shows, there is a smaller percentage of tables with small number of columns, with roughly 59% of the tables having 4 or more columns. This confirms the intuition that higher quality tables are more likely to have more number of columns, although there is still a significant number of tables with meaningful contents that have 3 or less columns.

Figure 3(c) shows the overall distribution of the number of rows in the whole corpus. Again, the majority of the tables are smaller ones, with roughly 78 million tables having under 20 rows, and roughly 1.5 million tables containing over 100 rows. Figure 3(d) shows the same statistics for tables with an overlap score over 20. Here again, the distribution of rows is clearly different from the whole corpus, with the majority of the tables having over 100 rows.

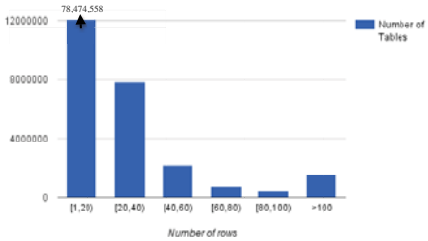
Next, we study the distribution of overlap scores over all tables and across different ontologies. Figure 4 shows the results (Schema.org results omitted for brevity). In all cases, the majority of tags have a score under 40, but there is a notable percentage of tags with a score above 100, i.e., the column has over 100 values shared with the set of labels of at least one type in the reference knowledge, a clear indication that the table is describing entities of that type. The main difference in the results across different ontologies is in the overall number of tags. With overlap score threshold of 20, there are 1,736,531 DBpedia Ontology tags, 542,178 Schema.org, 6,319,559 YAGO, 26,620,967 Freebase, and 865,718 Wikidata tags. The number of tags is a function of the size of the ontology in terms of number of classes and instances, but also the type system in the ontology. For example, Schema.org has only 43 classes resulting in an average of over 12,600 columns per each tag, but YAGO contains 368,447 classes which means an average of 17 columns per tag.



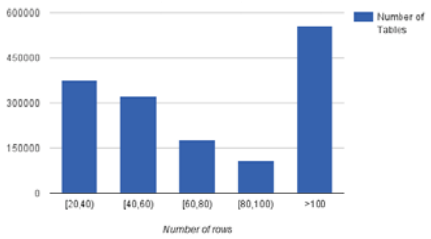
(a) Distribution of Number of Columns per Table



(b) Distribution of Number of Columns for Tables with 20+ Overlap Score



(c) Distribution of Number of Rows per Table



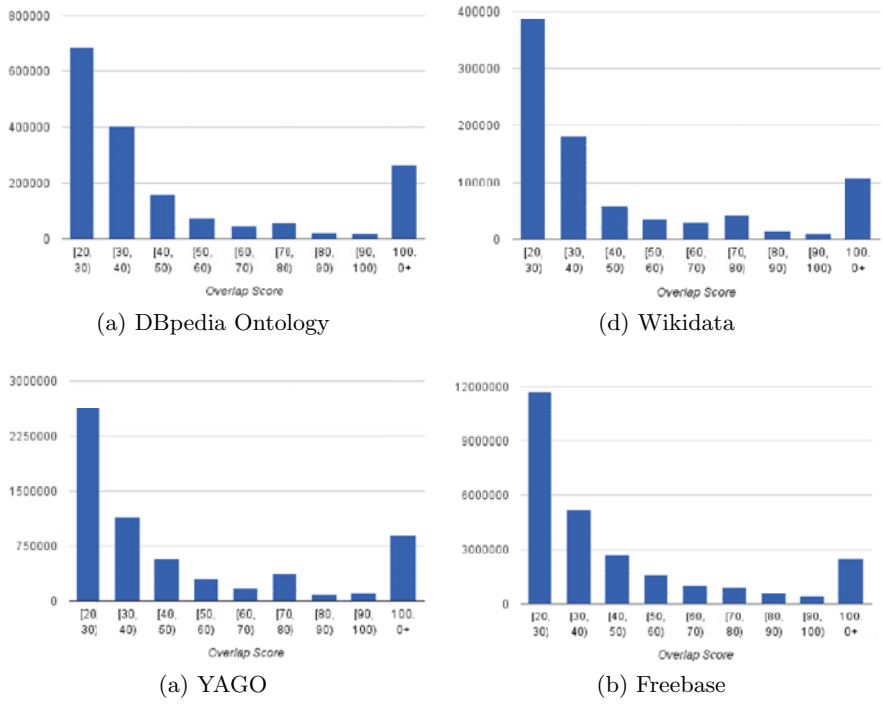
(d) Distribution of Number of Rows per Table with 20+ Overlap Score

Fig. 3. Distribution of Number of Rows and Columns

4 Distribution of Entity Types

We now present detailed statistics on the tags returned by the overlap similarity analysis described in Section 2. Going back to Figure 1 in Section 1, the word cloud figures are generated using the overlap analysis with the overlap threshold set to 20. The figure is then made using the top 150 most frequent tags in the output of the overlap analysis, with the size of each tag reflecting the number of columns annotated with that tag. The labels are derived either from the last portion of the class URI (for DBpedia and Freebase), or by looking up English class labels (for Wikidata). For example, "Person" in Figure 1(a) represents class `http://dbpedia.org/ontology/Person` whereas `music.recording` in Figure 1(c) represents `http://rdf.freebase.com/ns/music.recording`, and `chemical_compound` in Figure 1(d) represents `https://www.wikidata.org/wiki/Q11173` which has "chemical compound" as its English label.

In addition to the word cloud figures, Tables 1 and 2 show the top 20 most frequent tags in the output of our similarity analysis for each of the ontologies, along with their frequency in the output. From these results, it is clear that no single ontology on its own can provide the full picture of the types of entities that can be found on the Web tables. DBpedia ontology seem to have a better



(a) DBpedia Ontology (d) Wikidata

(a) YAGO (b) Freebase

Fig. 4. Distribution of Overlap Scores in Different Ontologies

coverage for person and place related entities, whereas YAGO has a large number of abstract classes being most frequent in the output. Schema.org provides a cleaner view over the small number of types it contains. Wikidata has a few surprising types on the top list, such as "commune of France". This may be due to a bias on the source on the number of editors contributing to entities under certain topics. Freebase clearly has a better coverage for media-related types, and the abundance of tags in music and media domain shows both the fact that there is a large number of tables in the Web tables corpus containing music and entertainment related contents, and that Freebase has a good coverage in this domain.

Finally, we examine a sample set of entity types across knowledge bases and see how many times they appear as a column tag in the overlap analysis output. Table 3 shows the results. Note that we have picked popular entity types that can easily be mapped manually. For example, Person entity type is represented by class `http://dbpedia.org/ontology/Person` in DBpedia, `http://dbpedia.org/class/yago/Person` in YAGO, `http://schema.org/Person` in Schema.org and

Table 1. Most Frequent Tags in DBpedia Ontology, YAGO, and Schema.org

DBpedia Ontology		YAGO		Schema.org	
Type	**Freq.**	**Type**	**Freq.**	**Type**	**Freq.**
Agent	242,410	PhysicalEntity	364,830	Person	186,332
Person	186,332	Object	349,139	Place	120,361
Place	120,361	YagoLegalActorGeo	344,487	CreativeWork	53,959
PopulatedPlace	112,647	Whole	230,667	Organization	50,509
Athlete	85,427	YagoLegalActor	226,633	Country	37,221
Settlement	60,219	YagoPerm.LocatedEntity	198,304	MusicGroup	22,926
ChemicalSubstance	57,519	CausalAgent	186,789	EducationalOrg.	12,159
ChemicalCompound	57,227	LivingThing	182,570	City	10,743
Work	53,959	Organism	182,569	CollegeOrUniversity	10,598
Organisation	50,509	Person	175,501	Movie	10,243
OfficeHolder	40,198	Abstraction	145,407	SportsTeam	9,594
Politician	39,121	LivingPeople	136,955	MusicAlbum	4,786
Country	37,221	YagoGeoEntity	120,433	Book	2,103
BaseballPlayer	30,301	Location	109,739	School	1,181
MotorsportRacer	26,293	Region	106,200	MusicRecording	1,166
RacingDriver	25,135	District	95,294	Product	1,130
Congressman	24,143	AdministrativeDistrict	92,808	TelevisionStation	1,037
MusicalWork	17,881	Group	85,668	StadiumOrArena	918
NascarDriver	16,766	Contestant	60,177	AdministrativeArea	896
Senator	15,087	Player	56,373	RadioStation	815

`http://rdf.freebase.com/ns/people.person` in Freebase. The numbers show a notable difference between the number of times these classes appear as column tags, showing a different coverage of instances across the knowledge bases. Freebase has by far the largest number of tags in these sample types. Even for the three ontologies that have the same instance data from DBpedia, there is a difference between the number of times they are used as a tag, showing that for example there are instances in DBpedia that have type Person in DBpedia ontology and Schema.org but not YAGO, and surprisingly, there are instances of Country class type in YAGO that are not marked as Country in DBpedia ontology or Schema.org.

5 Conclusion & Future Directions

In this paper, we presented the results of our study on understanding a large corpus of web tables through matching with public cross-domain knowledge bases. We focused on only one mechanism for understanding the corpus of tables, namely, tagging columns with entity types (classes) in knowledge bases. We believe that our study with its strict focus can provide new insights into the use of public cross-domain knowledge bases for similar analytics tasks. Our results clearly show the difference in size and coverage of domains in public cross-domain knowledge bases, and how they can affect the results of a large-scale analysis. Our results also show several issues in the Web Data Commons Web Tables corpus, such as the relatively large number of tables that contain very little or no meaningful contents.

Our immediate next step includes expanding this study to include other similarity measures and large-scale instance matching techniques [9]. Another interesting direction for future work is studying the use of domain-specific knowledge

Table 2. Most Frequent Tags in Wikidata and Freebase

Wikidata		Freebase	
Type	**Freq.**	**Type**	**Freq.**
Wikimedia_category	146,024	music.release_track	968,121
human	93,544	music.recording	964,906
chemical_compound	52,380	music.single	950,099
sovereign_state	34,681	location.location	532,053
country	22,030	people.person	475,472
determinator_for_. . . _occurrence	13,354	location.dated_location	460,766
city	12,823	location.statistical_region	458,643
commune_of_France	10,459	tv.tv_series_episode	440,985
taxon	10,127	location.citytown	409,315
landlocked_country	8,899	music.artist	390,458
island_nation	7,439	fictional_universe.fictional_character	372,820
republic	7,431	film.film_character	344,755
university	4,083	music.album	314,494
town	3,467	music.release	306,857
American_football_club	3,207	media_common.creative_work	304,231
band	3,024	media_common.cataloged_instance	297,875
municipality_of_Spain	2,950	type.content	269,216
comune_of_Italy	2,531	common.image	269,213
basketball_team	2,041	book.written_work	248,902
municipality_of_Germany	1,923	book.book	235,165

Table 3. Sample Entity Types and Their Frequency in Overlap Analysis Tags

Type	**DBpedia Ontology**	**YAGO**	**Schema.org**	**Wikidata**	**Freebase**
Person	186,332	175,501	186,332	93,544	475,472
Company	12,066	11,770	–	1,831	68,710
Location	120,361	109,739	120,36	–	532,053
Country	37,221	39,338	37,221	22,030	39,316
Film	10,243	9,080	10,243	348	175,460

bases to study the coverage of a certain domain in the corpus of Web Tables. For example, biomedical ontologies can be used in matching to discover healthcare related structured data on the Web.

The results reported in this paper may change after the reference knowledge sources or the corpus of tables are updated. Therefore, our plan is to maintain a website containing our latest results, along with the output of our analysis that can be used to build various search and discovery applications over the Web Tables corpus[1].

References

1. Google Web Tables. `http://research.google.com/tables`. [Online; accessed 29-04-2015].
2. Microsoft Excel Power Query. `http://office.microsoft.com/powerbi`. [Online; accessed 29-04-2015].
3. S. Balakrishnan, A. Y. Halevy, B. Harb, H. Lee, J. Madhavan, A. Rostamizadeh, W. Shen, K. Wilder, F. Wu, and C. Yu. Applying WebTables in Practice. In *CIDR*, 2015.

[1] For latest results, refer to our project page: `http://purl.org/net/webtables`.

4. C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, and S. Hellmann. DBpedia - A Crystallization Point for the Web of Data. *JWS*, 7(3):154–165, 2009.
5. K. D. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *SIGMOD*, pages 1247–1250, 2008.
6. M. J. Cafarella, A. Y. Halevy, D. Zhe Wang, E. Wu, and Y. Zhang. WebTables: Exploring the Power of Tables on the Web. *PVLDB*, 1(1):538–549, 2008.
7. B. B. Dalvi, W. W. Cohen, and J. Callan. WebSets: extracting sets of entities from the web using unsupervised information extraction. In *WSDM*, pages 243–252, 2012.
8. D. Deng, Y. Jiang, G. Li, J. Li, and C. Yu. Scalable Column Concept Determination for Web Tables Using Large Knowledge Bases. *PVLDB*, 6(13):1606–1617, 2013.
9. S. Duan, A. Fokoue, O. Hassanzadeh, A. Kementsietsidis, K. Srinivas, and M. J. Ward. Instance-Based Matching of Large Ontologies Using Locality-Sensitive Hashing. In *ISWC*, pages 49–64, 2012.
10. J. B. Ellis, A. Fokoue, O. Hassanzadeh, A. Kementsietsidis, K. Srinivas, and M. J. Ward. Exploring Big Data with Helix: Finding Needles in a Big Haystack. *SIGMOD Record*, 43(4):43–54, 2014.
11. F. Erxleben, M. Günther, M. Krötzsch, J. Mendez, and D. Vrandecic. Introducing Wikidata to the Linked Data Web. In *ISWC*, pages 50–65, 2014.
12. H. Mühleisen and C. Bizer. Web Data Commons - Extracting Structured Data from Two Large Web Corpora. 2012.
13. P. Ristoski, O. Lehmberg, R. Meusel, C. Bizer, A. Diete, N. Heist, S. Krstanovic, and T. A. Knller. Web Data Commons - Web Tables. `http://webdatacommons.org/webtables`. [Online; accessed 29-04-2015].
14. P. Ristoski, O. Lehmberg, H. Paulheim, and C. Bizer. Web Data Commons - English Subset of the Web Tables Corpus. `http://webdatacommons.org/webtables/englishTables.html`. [Online; accessed 29-04-2015].
15. P. Venetis, A. Y. Halevy, J. Madhavan, M. Pasca, W. Shen, F. Wu, G. Miao, and C. Wu. Recovering Semantics of Tables on the Web. *PVLDB*, 4(9):528–538, 2011.
16. D. Vrandecic and M. Krötzsch. Wikidata: a free collaborative knowledgebase. *Commun. ACM*, 57(10):78–85, 2014.

Combining Sum-Product Network and Noisy-Or Model for Ontology Matching

Weizhuo Li

Institute of Mathematics,Academy of Mathematics and Systems Science,
Chinese Academy of Sciences, Beijing, P. R. China
liweizhuo@amss.ac.cn

Abstract. Ontology matching is the key challenge to achieve semantic interoperability in building the Semantic Web. We present an alternative probabilistic scheme, called GMap, which combines the sum-product network and the noisy-or model. More precisely, we employ the sum-product network to encode the similarities based on individuals and disjointness axioms across ontologies and calculate the contributions by the maximum a posterior inference. The noisy-or model is used to encode the probabilistic matching rules, which are independent of each other as well as the value calculated by the sum-product network. Experiments show that GMap is competitive with many OAEI top-ranked systems. Futhermore, GMap, benefited from these two graphical models, can keep inference tractable in the whole matching process.

1 Introduction

Ontology matching is the process of finding relationships or correspondences between entities of different ontologies[5]. Many efforts have been conducted to automate the discovery in this process, e.g., incorporating more elaborate approaches including scaling strategies[3, 6], ontology repair techniques to ensure the alignment coherence[8], employing machine learning techniques[4], using external resources to increase the available knowledge for matching[2] and utilizing probabilistic graphical models to describe the related entities[1, 10, 11].

In this paper, we propose an alternative probabilistic schema, called GMap, based on two special graphical models—sum-product network(SPN) and noisy-or model. SPN is a directed acyclic graph with variables as leaves, sums and products as internal nodes, and weighted edges[12]. As it can keep inference tractable and describe the context-specific independence[12], we employ it to encode the similarities based on individuals and disjointness axioms and calculate the contributions by the maximum a posterior inference. Noisy-or model is a special kind of Bayesian Network[9]. When the factors are independent of each other, it is more suitable than other graphical models, specially in the inference efficiency[9]. Hence, we utilize it to encode the probabilistic matching rules. Thanks to the tractable inference of these special graphical models, GMap can keep inference tractable in the whole matching process. To evaluate GMap, we adopt the data sets from OAEI ontology matching campaign. Experimental results indicate that GMap is competitive with many OAEI top-ranked systems.

2 Methods

In this section, we briefly introduce our approach. Given two ontologies O_1 and O_2, we calculate the lexical similarity based on edit-distance, external lexicons and TFIDF[5]. Then, we employ SPN to encode the similarities based on individuals and disjointness axioms and calculate the contributions. After that, we utilize the noisy-or model to encode the probabilistic matching rules and the value calculated by SPN. With one-to-one constraint and crisscross strategy in the refine module, GMap obtains initial matches. The whole matching procedure is iterative. If it does not produce new matches, the matching is terminated.

2.1 Using SPN to encode individuals and disjointness axioms

In open world assumption, individuals or disjointness axioms are missing at times. Therefore, we define a special assignment—"$Unknown$" for the similarities based on these individuals and disjointness axioms.

For the similarity based on individuals, we employ the string equivalent to judge the equality of them. When we calculate the similarity of concepts based on individuals across ontologies, we regard individuals of each concept as a set and use Ochiai coefficient[1] to measure the value. We use a boundary t to divide the value into three assignments(i.e., 1, 0 and $Unknown$). Assignment 1(or 0) means that the pair matches(or mismatches). If the value ranges between 0 and t or the individuals of one concept are missing, the assignment is $Unknown$.

For the similarity based on disjointness axioms, we utilize these axioms and subsumption relations within ontologies and define some rules to determine its value. For example, x_1, y_1 and x_2 are concepts that come from O_1 and O_2. If x_1 matches x_2 and x_1 is disjoint with y_1, then y_1 is disjoint with x_2. The similarity also have three assignments. Assignment 1(or 0) means the pair mismatches(or overlaps). Otherwise, the similarity based on disjointness axioms is $Unknown$.

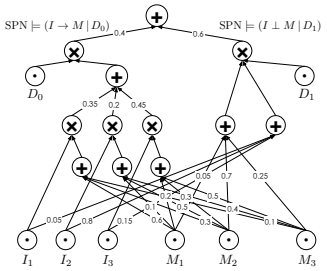


Fig. 1: The designed sum-product network

As shown in Figure 1, we designed a sum-product network S to encode above similarities and calculate the contributions, where M represents the contributions and leaves M_1, M_2, M_3 are indicators that comprise the assignments of M. All the indicators are binary-value. $M_1 = 1$(or $M_2 = 1$) means that the contributions are positive(or negative). If $M_3 = 1$, the contributions

[1] https://en.wikipedia.org/wiki/Cosine_similarity

are $Unknown$. Leaves I_1, I_2, I_3, D_0, D_1 are also binary-value indicators that correspond to the assignments of similarities based on individuals(I) and disjointness axioms(D). The concrete assignment metrics are listed in Table 1–2.

Table 1: Metric for Similarity D

Assignments	Indicators
$D = 1$	$D_0 = 0, D_1 = 1$
$D = 0$	$D_0 = 1, D_1 = 0$
$D = Unknown$	$D_0 = 1, D_1 = 1$

Table 2: Metric for Similarity I

Assignments	Indicators
$I = 1$	$I_1 = 1, I_2 = 0, I_3 = 0$
$I = 0$	$I_1 = 0, I_2 = 1, I_3 = 0$
$I = Unknown$	$I_1 = 0, I_2 = 0, I_3 = 1$

With the maximum a posterior(MAP) inference in SPN[12], we can obtain the contributions M. As the network S is complete and decomposable, the inference in S can be computed in time linear in the number of edges[7].

2.2 Using Noisy-Or model to encode probabilistic matching rules

We utilize probabilistic matching rules to describe the influences among the related pairs across ontologies and some of rules are listed in Table 3.

Table 3: The probabilistic matching rules among the related pairs

ID	Category	Probabilistic matching rules
R_1	class	two classes probably match if their fathers match
R_2	class	two classes probably match if their children match
R_3	class	two classes probably match if their siblings match
R_4	class	two classes about domain probably match if related objectproperties match and range of these property match
R_5	class	two classes about range probably match if related objectproperties match and domain of these properties match
R_6	class	two classes about domain probably match if related datapproperties match and value of these properties match

When we focus on calculating the matching probability of one pair, the matching rules are independent of each other as well as the value calculated by SPN. Therefore, we utilize the noisy-or model to encode them.

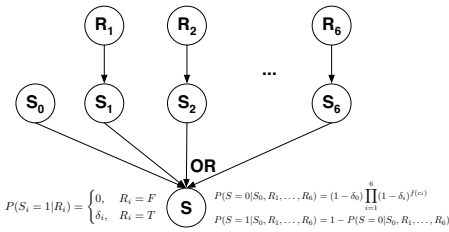


Fig. 2: The network structure of noisy-or model designed in GMap

Figure 2 shows the designed network, where R_i corresponds to the ith rule and S_i is the conditional probability depended on the condition of R_i. S_0 represents the SPN-based similarity that is a leak probability[9]. The matching

probability of one pair, $P(S = 1|S_0, R_1, ..., R_6)$, is calculated according to the formulas in the lower-right corner. c_i is the count of satisfied R_i and sigmoid function $f(c_i)$ is used to limit the upper bound of contribution of R_i. As the inference in the noisy-or model can be computed in time linear in size of nodes[9], GMap can keep inference tractable in the whole matching process.

3 Evaluation

To evaluate our approach[2], we adopt three tracks(i.e., Benchmark, Conference and Anatomy) from OAEI ontology matching campaign in 2014[3].

3.1 Comparing against the OAEI top-ranked systems

Table 4 shows a comparison of the matching quality of GMap and other OAEI top-ranked systems, which indicates that GMap is competitive with these promising existent systems. For Anatomy track, GMap does not concentrate on language techniques and it emphasizes one-to-one constraint. Both of them may cause a low alignment quality. In addition, all the top-ranked systems employ alignment debugging techniques, which is helpful to improve the quality of alignment. However, we do not employ these techniques in the current version.

Table 4: The comparison of GMap with the OAEI top-ranked systems

	Benchmark(Biblio)			Conference			Anatomy		
System	P	R	F	P	R	F	P	R	F
AML	**0.92**	0.4	0.55	**0.85**	0.64	**0.73**	**0.956**	**0.932**	**0.944**
LogMap	0.39	0.4	0.39	0.8	0.59	0.68	0.918	0.846	0.881
XMAP	1	0.4	0.57	0.87	0.49	0.63	0.94	0.85	0.893
CODI	n/a	n/a	n/a	0.74	0.57	0.64	0.967	0.827	0.891
GMap	0.63	**0.57**	**0.60**	0.67	**0.66**	0.66	0.930	0.802	0.862

3.2 Evaluating the contributions of these two graphical models

We separate SPN and the noisy-or model from GMap and evaluate their contributions respectively. As listed in Table 5, SPN is suitable to the matching task that the linguistic levels across ontologies are different and both of ontologies use same individuals to describe the concepts such as Biblio(201–210) in Benchmark track. Thanks to the contributions of individuals and disjointness axioms, SPN can improve the precision of GMap. When the structure information is very rich across the ontologies, the noisy-or model is able to discover some hidden matches with the existing matches and improve the recall such as in Anatomy track. However, if the ontology does not contain above features such as in Conference track, the improvement is not evident. Nevertheless, thanks to the complementary of these two graphical models to some extent, combining the sum-product network and the noisy-or model can improve the alignment quality as a whole.

[2] The software and results are available at https://github.com/liweizhuo001/GMap.
[3] http://oaei.ontologymatching.org/2014/

Table 5: The contributions of the sum-product network and the noisy-or model

	Biblio(201-210)			Conference			Anatomy		
System	P	R	F	P	R	F	P	R	F
string equivalent	0.680	0.402	0.505	0.8	0.43	0.56	**0.997**	0.622	0.766
lexical similarity(ls)	0.767	0.682	0.722	0.666	0.657	0.661	0.929	0.752	0.831
ls+spn	0.776	0.685	0.728	0.667	0.657	0.661	0.930	0.752	0.832
ls+noisy-or	0.782	0.701	0.739	**0.667**	**0.660**	**0.663**	0.937	0.772	0.847
ls+spn+noisy-or	**0.794**	**0.703**	**0.746**	**0.667**	**0.660**	**0.663**	0.930	**0.803**	**0.862**

4 Conclusion and Future Work

We have presented GMap, which is suitable for the matching task that many individuals and disjointness axioms are declared or the structure information is very rich. However, it still has a lot of room for improvement. For example, language techniques is essential to improve the quality of initial matches. In addition, dealing with alignment incoherent is also one of our future works.

Acknowledgments. This work was supported by the Natural Science Foundation of China (No. 61232015). Many thanks to Songmao Zhang, Qilin Sun and Yuanyuan Wang for their helpful discussion on the design and implementation of the GMap.

References

1. Albagli, S., Ben-Eliyahu-Zohary, R., Shimony, S.E.: Markov network based ontology matching. Journal of Computer and System Sciences 78(1), 105–118 (2012)
2. Zhang, S., Bodenreider, O.: Experience in aligning anatomical ontologies. International journal on Semantic Web and information systems 3(2), 1–26 (2007)
3. Djeddi, W.E, Khadir, M.T.: XMAP: a novel structural approach for alignment of OWL-full ontologies. In: Proc. of Machine and Web Intelligence(ICMWI). pp. 368–373 (2010)
4. Doan, A.H, Madhavan, J., Dhamankar, R., et al.: Learning to match ontologies on the semantic web. The VLDB Journal 12(4), 303–319 (2003)
5. Euzenat, J., Shvaiko, P.: Ontology Matching(2nd Edition). Springer (2013)
6. Faria, D., Pesquita, C., Santos, E., et al.: The agreementmakerlight ontology matching system In: 2013 OTM Conferences. pp. 527–541 (2013)
7. Gens, R., Pedro, D.: Learning the structure of sum-product networks. In: Proc. of International Conference on Machine Learning(ICML). pp. 873–880 (2013)
8. Jimenez-Ruiz, E., Grau, B.C.: LogMap: Logic-based and scalable ontology matching. In: Proc. of International Semantic Web Conference(ISWC). pp.273–288 (2011)
9. Koller, D., Friedman, N.: Probabilistic Graphical Models. MIT press (2009)
10. Mitra, P., Noy, N.F., Jaiswal, A.R. OMEN: A probabilistic ontology mapping tool. In: Proc. of International Semantic Web Conference(ISWC). pp. 537–547 (2005)
11. Niepert, M., Noessner, J., Meilicke, C., Stuckenschmidt, H.: Probabilistic-logical web data integration. Reasoning Web. pp. 504–533 (2011)
12. Poon, H., Domingos, P.: Sum-product networks: A new deep architecture. In: Proc of International Conference on Computer Vision Workshops(ICCV Workshops). pp. 689–690 (2011)

Towards Combining Ontology Matchers via Anomaly Detection

Alexander C. Müller and Heiko Paulheim

University of Mannheim, Germany
Research Group Data and Web Science
heiko@informatik.uni-mannheim.de,alexanda@mail.uni-mannheim.de

Abstract. In ontology alignment, there is no single best performing matching algorithm for every matching problem. Thus, most modern matching systems combine several *base matchers* and aggregate their results into a final alignment. This combination is often based on simple voting or averaging, or uses existing matching problems for learning a combination policy in a *supervised* setting. In this paper, we present the *COMMAND* matching system, an *unsupervised* method for combining base matchers, which uses anomaly detection to produce an alignment from the results delivered by several base matchers. The basic idea of our approach is that in a large set of potential mapping candidates, the scarce actual mappings should be visible as anomalies against the majority of non-mappings. The approach is evaluated on different OAEI datasets and shows a competitive performance with state-of-the-art systems.

Keywords: Ontology Alignment, Anomaly Detection, Outlier Detection, Matcher Aggregation, Matcher Selection

1 Introduction

In ontology matching, there is only rarely a *one size fits all* solution. Ontology matching problems differ along many dimensions, so that a matching system that performs well on one dataset does not necessarily deliver good results on another one. To overcome this problem, many ontology matching tools combine the results of various base matchers, i.e., individual matching strategies. However, this approach gives way to a new problem, i.e., how to *combine* the results of the base matchers in a way that the combination suits the problem at hand [7]. Solutions proposed in the past range from simple voting to supervised learning.

In this paper, we propose to use *anomaly* or *outlier detection* for the problem of matcher combination. Anomaly detection is the task of finding those data points in a data set that deviate from the majority of the data [1]. The underlying assumption is that given a large set of mapping candidates (e.g., the cross product of ontology elements from the ontologies at hand), the *actual* mappings (which are just a few) should stand out in one way or the other. Thus, it should be possible to discover them using anomaly detection methods. We show that it is possible to build a competitive matching system combining the results of more than 25 base matchers using anomaly detection.

2 Approach

COMMAND is a novel approach for dynamically selecting and combining ontology matchers via anomaly detection. The overall architecture is depicted in Fig. 1. The platform was implemented in Scala, the code is available on github under an open-source license.[1]

2.1 Base Matching and Matcher Selection

First, all base matchers that are based on local information of each ontology entity are executed. The entities of the target and source ontology are matched in a pair-wise fashion. This step matches *Classes*, *DataProperties* and *ObjectProperties* pairwise and independently.

After this the first *feature vector* is analyzed and an uncorrelated feature subset is extracted. The results of those uncorrelated matchers are used as the input similarities for the *structural matchers*.

The result of the *structural matchers* is joined with the element level matcher result to create a *feature vector*. Since some of the features might be redundant or not vary in their values and thus do not contribute to the final matching, we remove results with little variation, correlated results, and also support PCA for computing meaningful linear combinations of base matcher results.

The current version of COMMAND implements a large variety of element and structure level techniques. Those encompass 16 string similarity metrics, five external metrics based on WordNet and corpus linguistics, and five structural matching techniques, such as similarity flooding.

2.2 Aggregation by Anomaly Detection

The next step is the aggregation of the base matcher results into a final matching score for all correspondences. We perform this step by detecting outlying datapoints in the *feature vector space*, and using this score as a measure of similarity. The anomaly analysis and score normalization are performed separately for classes, data properties, and object properties.

To compute outlier scores, we apply *anomaly analysis techniques* on the feature vector representations. In this paper, we use three different techniques: A *k-nearest-neighbor based method (KNN)* that computes the anomaly score of a data point based on the average euclidean distances[2] to its nearest neighbors, a cluster-based method that calculates the unweighted *cluster-based local anomaly factor (CBLOF)* based on a given clustering scheme produced by an arbitrary clustering algorithm [5], and the *Replicator Neural Networks (RNN)* method, which trains a neural network capturing the patterns in the data, and identifies those data points not adhering to those patterns [4].

[1] https://github.com/dwslab/COMMAND

[2] Note that since we expect all base matcher scores to fall in a $[0; 1]$ interval, using geometrical distance measures in that space is feasible.

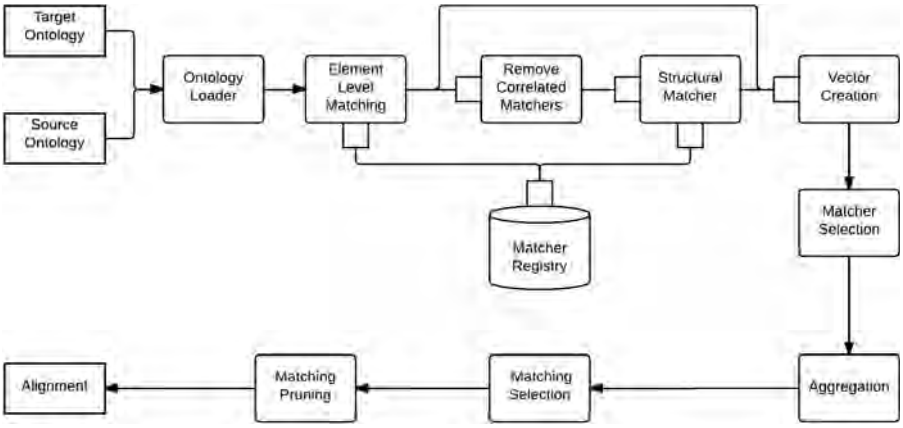


Fig. 1. Overview of the COMMAND pipeline

2.3 Matching Selection and Repair

The result of the previous step is a set of candidates, which does not necessarily form a semantically coherent mapping. After applying a threshold to the results of classes, data and object properties, the mapping may be refined by the *Hungarian method*, a *greedy selection*, or a *fuzzy greedy selection* [2]. Furthermore, logical consistency may be ensured by running the *ALCOMO* mapping post-processing system [6].

3 Evaluation

To evaluate the COMMAND approach, we use the *benchmark*, *conference*, and *anatomy* of the Ontology Alignment Evaluation Initiative (OAEI) 2014 [3].

We compare the results of COMMAND to three baselines. *Single best global* refers to the single base matcher that performs best on the given test case (i.e., conference, benchmark, and anatomy), using the optimal global threshold. *Majority vote* performs a voting across all base matchers, again using the best global threshold. *Single best local* selects the best base matcher for each problem.[3]

Furthermore, we compare COMMAND to the contestants of the OAEI 2014 initiative. To make that comparison fair, we use one global parameter set for each variant across all three OAEI datasets, instead of per dataset settings.

Tables 1, 2, and 3 depict the results of COMMAND on the OEAI datasets, once with and once without the use of ALCOMO. For anatomy, we restrict ourselves to the CBLOF variant and a subset of eight element-level matchers due to reasons of runtime. Except for the *Single best local* baseline (which is informative and not a baseline that can actually be implemented), COMMAND outperforms all baselines. When comparing COMMAND to the results of OAEI

[3] Note that in practice, it would not be possible to implement a matcher like *Single best local*. We only report it for informative purposes.

Table 1. Results on the OAEI biblio benchmark dataset. The table reports macro average recall, precision, and F-measure, with micro average values in parantheses.

Approach	without ALCOMO			with ALCOMO		
	Precision	Recall	F1	Precision	Recall	F1
Single best global	.754 (.733)	.557 (.521)	.641 (.609)	.779 (.761)	.548 (.521)	.644 (.619)
Majority vote	.510 (.472)	.570 (.544)	.538 (.505)	.524 (.487)	.463 (.443)	.491 (.464)
Single best local	.788 (.718)	.632 (.616)	.702 (.663)	.835 (.798)	.610 (.584)	.705 (.674)
CBLOF + PCA	.833 (.983)	.444 (.470)	.579 (.636)	.832 (.981)	.432 (.457)	.568 (.624)
CBLOF + RC	.844 (.982)	.466 (.461)	.600 (.627)	.844 (.982)	.457 (.449)	.593 (.617)
k-NN + PCA	.868 (.977)	.547 (.550)	**.672** (.704)	.871 (.975)	.480 (.459)	**.619 (.624)**
k-NN + RC	.847 (.967)	.549 (.556)	.666 (**.706**)	.835 (.984)	.463 (.442)	.596 (.610)
RNN + PCA	.881 (.991)	.466 (.443)	.610 (.612)	.859 (.965)	.324 (.253)	.470 (.401)
RNN + RC	.877 (.988)	.470 (.448)	.612 (.616)	.877 (.987)	.471 (.450)	.613 (.618)

Table 2. Results on the OAEI conference dataset. The table reports macro average recall, precision, and F-measure, with micro average values in parantheses.

Approach	without ALCOMO			with ALCOMO		
	Precision	Recall	F1	Precision	Recall	F1
Single best global	.641 (.784)	.591 (.611)	.615 (.687)	.640 (.783)	.591 (.611)	.615 (.686)
Majority vote	.874 (.949)	.537 (.552)	.665 (.698)	.874 (.949)	.537 (.552)	.665 (.698)
Single best local	.651 (.795)	.602 (.625)	.626 (.700)	.650 (.793)	.602 (.625)	.625 (.699)
CBLOF + PCA	.693 (.678)	.636 (.613)	.663 **(.644)**	.737 (.715)	.625 (.600)	**.676 (.652)**
CBLOF + RC	.702 (.693)	.607 (.577)	**.651** (.630)	.761 (.752)	.588 (.557)	.663 (.640)
k-NN + PCA	.718 (.712)	.572 (.534)	.636 (.610)	.797 (.782)	.557 (.518)	.656 (.623)
k-NN + RC	.710 (.702)	.574 (.541)	.635 (.611)	.781 (.769)	.530 (.492)	.631 (.600)
RNN + PCA	.829 (.815)	.528 (.492)	.645 (.613)	.748 (.699)	.617 (.587)	**.676** (.638)
RNN + RC	.820 (.805)	.527 (.489)	.641 (.608)	.819 (.804)	.524 (.485)	.639 (.605)

2014, we can find that the system, using CBLOF and PCA, and alignment repair with ALCOMO, would score on rank on a shared fifth rank (with XMap2) for the benchmark track, on rank four for the conference track (between LogMap-C and XMap), and on rank six (between LogMap-C and MaasMatch) for the anatomy track.

The runtime of *COMMAND* is assessed by measuring the time of a complete end-to-end pipeline execution. The general time complexity of *COMMAND* is quadratic to the size of the input ontologies. Additionally, the time consumption of the individual steps is measured. The results are depicted in table 4.

4 Conclusion and Outlook

In this paper, we have introduced a novel approach using anomaly detection for combining the results of different ontology matchers into a final aggregated matching score.

Overall, *COMMAND* performs an efficient *matcher selection* that only considers matchers that contribute to the final result, and uses *anomaly detection* as an unsupervised method for aggregating base matcher results. It is superior

Table 3. Results on the OAEI anatomy dataset.

Approach	without ALCOMO			with ALCOMO		
	Precision	Recall	F1	Precision	Recall	F1
Single best local/global	.920	.773	.840	.918	.740	**.820**
Majority vote	.932	.606	.735	.931	.597	.727
CBLOF + PCA	.892	.728	**.801**	.911	.741	.817
CBLOF + RC	.839	.664	.742	.832	.725	.775

Table 4. Average runtime in seconds of COMMAND

Dataset	Ø total	Ø t vector creation	Ø t aggregation	Ø t extraction
Conference	69.267	53.580	15.683	0.004
Benchmarks	52.880	44.026	8.850	0.004
Anatomy	18,746.510	11,595.601	5,922.478	1,228.431

to a simple majority vote baseline and performs in the range of state of the art matching tools. Furthermore, the possibility to use principal component analysis for feature space transformation also allows for implicitly computing relevant linear combinations of matcher scores.

The evaluation has been carried out on three OAEI datasets. For *conference* and *benchmarks*, the system achieved competitive performances in comparison to other OAEI participants. The results on the *anatomy* track showed that, since only a reduced configuration could be used with sub-optimal results, that more memory-efficient implementations are still required for fully exploiting the capabilities of COMMAND.

Furthermore future work will include the inclusion of other anomaly detection approaches, like angle-based methods, as well as other score normalization methods.

References

1. Chandola, V., Banerjee, A., Kumar, V.: Anomaly detection: A survey. ACM Computing Surveys (CSUR) 41(3) (2009)
2. Do, H.H., Rahm, E.: Coma: A system for flexible combination of schema matching approaches. In: Proceedings of the 28th International Conference on Very Large Data Bases. pp. 610–621. VLDB '02, VLDB Endowment (2002)
3. Dragisic, Z.e.a.: Results of theontology alignment evaluation initiative 2014. In: International Workshop on Ontology Matching. pp. 61–104 (2014)
4. Hawkins, S., He, H., Williams, G., Baxter, R.: Outlier detection using replicator neural networks. In: Data warehousing and knowledge discovery, pp. 170–180. Springer (2002)
5. He, Z., Xu, X., Deng, S.: Discovering cluster-based local outliers. Pattern Recognition Letters 24(9), 1641–1650 (2003)
6. Meilicke, C.: Alignment incoherence in ontology matching. Ph.D. thesis (2011)
7. Shvaiko, P., Euzenat, J.: Ontology matching: State of the art and future challenges. Knowledge and Data Engineering, IEEE Transactions on 25(1), 158–176 (Jan 2013)

User Involvement in Ontology Matching Using an Online Active Learning Approach

Booma Sowkarthiga Balasubramani, Aynaz Taheri, and Isabel F. Cruz

ADVIS Lab
Department of Computer Science
University of Illinois at Chicago
{bbalas3,ataher2,ifcruz}@uic.edu

Abstract. We propose a semi-automatic ontology matching system using a hybrid active learning and online learning approach. Following the former paradigm, those mappings whose validation is estimated to lead to greater quality gain are selected for user validation, a process that occurs in each iteration, following the online learning paradigm. Experimental results demonstrate the effectiveness of our approach.

1 Introduction

The result of performing ontology matching is a set of mappings between concepts in the *source ontology* and concepts in the *target ontology*. This set is called an *alignment*. The *reference alignment* or *gold standard* is (an approximation of) the set of correct and complete mappings built by domain experts. We consider a semi-automatic ontology matching approach, whereby the mappings are first determined using automatic ontology matching methods, which we call *matchers*, followed by user validation.

We use six of the matchers of the AgreementMaker ontology matching system [3], including the Linear Weighted Combination (LWC) matcher, which performs a weighted combination of the results of the other five matchers, using weights that are automatically determined using a quality metric [4].

We train a classifier and modify the weights of the LWC matcher using an iterative approach, following the on-line learning paradigm. At each iteration, user validation is sought for those candidate mappings that can potentially contribute the most to the quality of the final alignment, following the active learning paradigm. The process continues until there is no significant improvement in F-Measure. We describe this process in Section 2. Experimental results are obtained using the ontology sets from the Ontology Alignment Evaluation Initiative (OAEI) and comparison is made with the results of other systems in Section 3. We discuss related work in Section 4, and conclude with Section 5.

2 Proposed System

After the source and target ontologies are loaded into AgreementMaker, the following steps are executed in sequence:

Automatic matching algorithms execution The following matchers are executed individually and their results are stored in the corresponding similarity matrices: the Advanced Similarity Matcher (ASM) [5], the Parametric String-based

Matcher (PSM) [4], the Lexical Similarity Matcher (LSM) [5], the Vector-based Multi-word Matcher (VMM) [4], and the Base Similarity Matcher (BSM) [5].

Linear weighted combination The Linear Weight Combination (LWC) matcher [6] linearly combines the similarity matrices of the other five automatic matchers using weights determined by the local confidence quality metric, which estimates the quality of the scores produced by each matcher. The new score for each mapping is stored in the LWC matrix. It is up to the selection phase to output only those mappings that are in the final alignment, taking into account the desired cardinality of the mappings (e.g., one-to-one) [4].

Candidate mapping selection Candidate mappings to be presented to the users for validation are based on the combination of the following three criteria: (1) Disagreement-based Top-k Mapping [6], which measures the level of similarity among the five scores, one for each of the matchers considered. If the matchers mostly agree on the scores, then the disagreement is low, but it is high when the matchers disagree on the scores; (2) Cross Count Quality (CCQ), which counts, for a score, the number of non-zero scores in the row and column of that score in the LWC matrix [2]. The count is normalized by the maximum sum of the scores per column and row in the whole matrix; (3) Similarity Score Definiteness (SSD), which is a quality metric that ranks mappings in increasing order of their score [2]. It evaluates how close the score associated with a mapping is to the maximum and minimum possible scores (1 and 0).

User validation The result of this step is a label that has value 1 if the mapping is correct and 0 if the mapping is incorrect. For each iteration, users validate a set of candidate mappings. The validation of each mapping is called an *interaction* by others [7]. There can be any number of interactions per iteration, that is, users can be presented with any number of mappings to validate at a time.

Classification We use a logistic regression classifier, which considers the parametric distribution $P(Y|X)$ where Y is the discrete-valued user label (1 or 0) and the feature vector $X = \langle X_1, \ldots, X_n \rangle$ is the signature vector [6] with n scores computed for a mapping by n individual matchers, and estimates the parameter that is the vector of weights $W = \langle w_1, \ldots, w_n \rangle$ of the LWC matcher. The logistic regression model is based on the following probabilities:

$$P(Y = 1|X) = \frac{1}{1 + e^{w_0 + \sum_{i=1}^{n} w_i X_i}}, P(Y = 0|X) = \frac{e^{w_0 + \sum_{i=1}^{n} w_i X_i}}{1 + e^{w_0 + \sum_{i=1}^{n} w_i X_i}}$$

W is updated during the iterative process by taking the partial derivative of the log likelihood function with respect to each component, w_i. The recursive rule for the update is as follows, where α is the learning rate that determines how fast or slow the weights will converge to their optimal values [10]:

$$W \leftarrow W + \alpha \sum_{i=1}^{m} X^i (Y^i - g(W^T X^i))$$

3 Experimental Evaluation

We use the 2014 OAEI Conference Track ontology sets and their reference alignments to simulate the user validation. The baseline is the F-Measure obtained

automatically by the AgreementMaker matchers. Table 1 depicts the average F-Measure after 20 iterations using the three candidate selection criteria individually or in combination with one another. The top performer is the Disagreement-based Top-k Mapping Selection criteria.

	1	2	3	4	5	6	7
Candidate Mapping Selection Strategy	48.08	52.45	60.43	51.42	48.91	52.47	53.18
Baseline (Before User Feedback)	51.8	51.8	51.8	51.8	51.8	51.8	51.8

Strategies: 1. CCQ 2. SSD 3. Disagreement 4. CCQ + SSD 5. CCQ + Disagreement 6. SSD + Disagreement 7. CCQ + SSD + Disagreement

Table 1: Average F-Measure for 20 iterations (123 interactions/iteration).

Matcher	F-Measure with User Feedback	F-Measure w/o User Feedback	F-Measure gain	Relative Number of Interactions
AML	0.801	0.730	0.071	0.497
LogMap	0.729	0.680	0.049	0.391
HerTUDA	0.582	0.600	-0.018	0.996
WeSeE	0.473	0.610	-0.137	0.447
Our Approach	0.604	0.518	0.086	0.470

Table 2: Comparison with the 2014 OAEI Interactive Track results.

Our approach has an average F-Measure gain of 8.6% and an average F-Measure of 60.4%. This is a considerable improvement as we started from an average F-Measure of 51.8%, which was obtained using the automatic matchers along with LWC. Table 2 compares our results with those obtained by other systems that participated in the 2014 OAEI Interactive Track. It performs better than HerTUDA and WeSeE (with F-Measure values of 58.2% and 47.3%, respectively). The F-Measure gain of AML [9] is 7.1% and of LogMap is 4.6%, therefore our approach has the highest F-Measure gain. The table also shows the relative number of interactions, which is the average number of interactions per pair of ontologies divided by the size of the reference alignment for that pair. Our approach shows better improvement in F-Measure with fewer number of interactions when compared to AML that has the highest F-Measure.

Figure 1 shows the effect of the total number of interactions on the F-Measure in our approach. Here, the total number of interactions represent the sum of the number of interactions in each of the 21 reference alignments in the Conference Track dataset (one for each pair of ontologies) up to 123 interactions. The Disagreement-based Top-k Mapping Selection performs better than the other candidate selection strategies. SSD and the combination of SSD+CCQ+ Disagreement have the next highest average F-Measure.

4 Comparison with Related Work

We divide previous work into two categories depending on whether feedback from single or multiple users is considered.

Single user A previous approach that uses AgreementMaker performs updates in the LWC matrix based on user feedback [6], but does not use a classifier to adjust

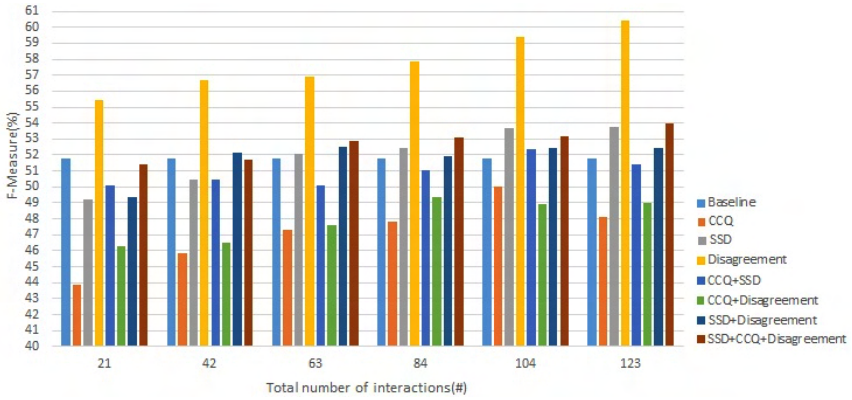


Fig. 1: F-Measure gain as a function of the number of interactions.

the LWC weights. Another method uses logistic regression to learn an optimal combination of both lexical and structural similarity metrics [8]. Compared to our approach, it uses different similarity metrics, candidate selection strategies, and techniques to customize weights for different matching strategies. Another system aggregates similarity measures with the help of self-organizing maps and incorporates user feedback for refining self-organizing map outcomes [11]. There is an active learning approach where the user validation is propagated according to the ontology structure [13]. Another approach makes use of the parameterization of matchers [12]. It uses example mappings to automatically determine a suitable parameter setting for each matcher, based on those examples. However, in our approach, the LWC uses five of the already existing matchers with the same configuration as in AgreementMaker.

Multiple users We discuss two approaches. The first one uses a pay-as-you-go approach and propagates the (possibly faulty) user validation input to similar mappings [2]. In the second approach, a multi-user feedback method that attempts to maximize the benefits that can be drawn from user feedback, by managing it as a first class citizen [1]. None of these approaches uses a classifier.

5 Conclusions and Future Work

In this paper, we have proposed an effective semi-automatic ontology matching approach that combines active learning with online learning. Our experimental evaluation demonstrate that a considerable improvement in F-Measure can be achieved over the base case. Clearly, a combination of user feedback with learning is fertile ground for future research, where the scalability of the methods to large and very large ontologies and the use of a variety of classifiers and of candidate selection strategies would be some of the topics to investigate.

Acknowledgments

This research was partially supported by NSF Awards IIS-1143926, IIS-1213013, and CCF-1331800.

References

1. Belhajjame, K., Paton, N.W., Fernandes, A.A.A., Hedeler, C., Embury, S.M.: User Feedback as a First Class Citizen in Information Integration Systems. In: CIDR Conference on Innovative Data Systems Research. pp. 175–183 (2011)
2. Cruz, I.F., Loprete, F., Palmonari, M., Stroe, C., Taheri, A.: Pay-As-You-Go Multi-User Feedback Model for Ontology Matching. In: International Conference on Knowledge Engineering and Knowledge Management (EKAW), pp. 80–96. Springer (2014)
3. Cruz, I.F., Palandri Antonelli, F., Stroe, C.: AgreementMaker: Efficient Matching for Large Real-World Schemas and Ontologies. PVLDB 2(2), 1586–1589 (2009)
4. Cruz, I.F., Palandri Antonelli, F., Stroe, C.: Efficient Selection of Mappings and Automatic Quality-driven Combination of Matching Methods. In: ISWC International Workshop on Ontology Matching (OM). CEUR Workshop Proceedings, vol. 551, pp. 49–60 (2009)
5. Cruz, I.F., Stroe, C., Caci, M., Caimi, F., Palmonari, M., Palandri Antonelli, F., Keles, U.C.: Using AgreementMaker to Align Ontologies for OAEI 2010. In: ISWC International Workshop on Ontology Matching (OM). CEUR Workshop Proceedings, vol. 689, pp. 118–125 (2010)
6. Cruz, I.F., Stroe, C., Palmonari, M.: Interactive User Feedback in Ontology Matching Using Signature Vectors. In: IEEE International Conference on Data Engineering (ICDE). pp. 1321–1324 (2012)
7. Dragisic, Z., Eckert, K., Euzenat, J., Faria, D., Ferrara, A., Granada, R., Ivanova, V., Jiménez-Ruiz, E., Kempf, A.O., Lambrix, P., Montanelli, S., Paulheim, H., Ritze, D., Shvaiko, P., Solimando, A., dos Santos, C.T., Zamazal, O., Grau, B.C.: Results of the Ontology Alignment Evaluation Initiative 2014. In: ISWC International Workshop on Ontology Matching (OM). pp. 61–104. CEUR Workshop Proceedings (2014)
8. Duan, S., Fokoue, A., Srinivas, K.: One Size Does Not Fit All: Customizing Ontology Alignment Using User Feedback. In: International Semantic Web Conference (ISWC). Lecture Notes in Computer Science, vol. 6496, pp. 177–192. Springer (2010)
9. Faria, D., Pesquita, C., Santos, E., Palmonari, M., Cruz, I.F., Couto, F.M.: The AgreementMakerLight Ontology Matching System. In: International Conference on Ontologies, DataBases, and Applications of Semantics (ODBASE). pp. 527–541. Springer (2013)
10. Halloran, J.: Classification: Naive Bayes vs Logistic Regression. Tech. rep., University of Hawaii at Manoa EE 645 (2009)
11. Jirkovskỳ, V., Ichise, R.: Mapsom: User Involvement in Ontology Matching. In: Joint International Semantic Technology Conference (JIST), pp. 348–363. Springer (2014)
12. Ritze, D., Paulheim, H.: Towards an Automatic Parameterization of Ontology Matching Tools Based on Example Mappings. In: ISWC International Workshop on Ontology Matching (OM). pp. 37–48 (2011)
13. Shi, F., Li, J., Tang, J., Xie, G., Li, H.: Actively Learning Ontology Matching via User Interaction. In: International Semantic Web Conference (ISWC). Lecture Notes in Computer Science, vol. 5823, pp. 585–600. Springer (2009)

ADOM: Arabic Dataset for Evaluating Arabic and Cross-lingual Ontology Alignment Systems

Abderrahmane Khiat[1], Moussa Benaissa[1], and Ernesto Jiménez-Ruiz[2]

[1]LITIO Laboratory, University of Oran1 Ahmed Ben Bella, Oran, Algeria
[2]Department of Computer Science, University of Oxford, United Kingdom

Abstract. In this paper, we present ADOM, a dataset in Arabic language describing the conference domain. This dataset was created for two purposes (1) analysis of the behavior of matchers specially designed for Arabic language, (2) integration with the multifarm dataset of the Ontology Alignment Evaluation Initiative (OAEI). The multifarm track evaluates the ability of matching systems to deal with ontologies described in different natural languages. We have tested the ADOM dataset with the LogMap ontology matching system. The experiment shows that the ADOM dataset works correctly for the task of evaluating cross multilingual ontology alignment systems.

1 Introduction

Ontology alignment is defined as the identification process of semantic correspondences between entities of different ontologies in order to ensure the semantic interoperability [1]. However, the automatic identification of correspondences between ontologies is very difficult due to (a) their conceptual divergence [8], and (b) to the use of different naming conventions or languages. In the literature there are several systems that deal with the (semi) automatic alignment of ontologies [1, 12, 11]. These systems are (typically) primarily based on the lexical similarity of the entity labels. Matching ontologies in different languages is challenging due to misinterpretations during the translation process. Ontologies in Arabic language brings even more challenges due to special features of the language. Among the reasons that make ontology alignment in Arabic language very difficult we can quote [6]:

1. The Arabic script (no short vowels and no capitalization).
2. Explosion of ambiguity (in average 2.3 per word in other languages to 19.2 in Arabic) by Buckwalter (2004) [5].
3. Complex word structure, for example the sentence ورأيتهم can be translated in English language as and I saw them.
4. The problem of Normalization, for example آ ، إ ، أ ، ا → ا i.e. losing distinction أن ، إن ، آن
5. The Arabic language is one of the pro-drop languages, i.e. languages that allow speakers to omit certain classes of pronouns

Table 1: Top systems in the multifarm track

OAEI	Top Systems	Precision	F-measure	Recall
2012	YAM++	0.50	0.40	0.36
2013	YAM++	0.51	0.40	0.36
2014	AML	0.57	0.54	0.53
2014	LogMap	0.80	0.40	0.28
2014	XMap	0.31	0.35	0.43

In this paper, we present ADOM, a dataset in Arabic language describing the conference domain. We have created this dataset by translating and improving all ontologies of the conference track [13] of the OAEI campaign. We summarize below the objectives of the developed dataset: (1) Analysis and evaluation of the behaviour of matchers designed for Arabic language. Here, the real questions are: (a) could the state of the art systems handle efficiently the ontologies described in Arabic language? (b) Are external knowledge resources for Arabic language available such as WordNet? (2) Integration with the multifarm track [14] of the OAEI campaign.[1] The multifarm track evaluates the ability of matching systems to deal with ontologies described in different natural languages. The question here, concerns to the performance of the translator used to align multilingual ontologies?

The rest of the paper is organized as follows. First, in Section 2, we discuss the top systems that participated in the last editions of the multifarm track. In section 3 we describe the ADOM dataset. Section 4 contains the experiment results. Finally, some concluding remarks and future work are presented in Section 5.

2 Related Work

In this section we discuss the main ontology matching systems that have participated in the multifarm track. Most of such systems use a translation tool to deal with the cross-lingual ontology alignment. The XMap system [2] uses an automatic translation for obtaining correct matching pairs in multilingual ontology matching. The translation is done by querying Microsoft Translator for the full name. The AML system [4] uses an automatic translation module based on Microsoft Translator. The translation is done by querying Microsoft Translator for the full name (rather than word-by-word). To improve performance, AML stores locally all translation results in dictionary files, and queries the Translator only when no stored translation is found. The LogMap system [10] that participated in the OAEI 2014 campaign used a multilingual module based on Google translate [3]; however the new version of the LogMap system uses both Microsoft and Google translator APIs [9]. The YAM++ system [7] uses a multilingual translator based on Microsoft Bing to translate the annotations to English. Table 1 summarizes the results of the top systems in the multifarm track.

[1] ADOM has already been integrated within the OAEI 2015 multifarm dataset: `http://oaei.ontologymatching.org/2015/multifarm/index.html`

3 The ADOM Dataset

The dataset is constituted of seven ontologies in Arabic language. These ontologies describe the conference domain and are based on the ontologies of the OAEI conference track [13]. We justify the proposal of our dataset by the following points: (1) The OAEI campaign, which is the most known evaluation campaign for testing the performance of ontology matching systems, lacked a test case involving ontologies in Arabic language. (2) To the best of our knowledge, no such dataset exists yet in Arabic language.[2] (3) Furthermore, there are several contexts such as Web information retrieval where the ontology matching systems are needed both in inter-multilingual ontologies and intra-Arabic ontologies.

We have developed our dataset relying on the conference and multifarm tracks of the OAEI. In order to develop the Arabic ontologies and reference alignments for the ADOM dataset we proceeded as follows.

3.1 Step 1: Translation of Ontology Entities

In this step, we have identified the concepts, object and data-type properties of the ontologies, for example we can list the concept "البحث (paper)", data-type property "لديه اسم (has name) and object property "لديه موقع على رابط (has website at URL), etc. We have semi-automatically translated the ontologies in English and French by considering the context of the ontologies (i.e., the conference domain). For example, if we translate simply the concept "paper" we get "ورقة" in Arabic language but "ورقة" is not the correct concept if we consider the context of conference and some information from conference websites in Arabic language. Then the correct concept of "paper" becomes "البحث".

3.2 Step 2: Generation of Reference Alignments

We have reused the available reference alignments among the ontologies in the multifarm track to generate the new reference alignments for ADOM. For example, in the reference alignment for ontologies in Arabic language, we can list the concept "الحدث (event)" of the ontology Confof is equivalent to the concept "نشاط (activity)" of the ontology Iasted. In the reference alignment for ontologies in Arabic and French languages, we can list the concept "éditeur (Editor)" of the ontology conference is equivalent to the concept "المحرر" of the ontology Cmt.

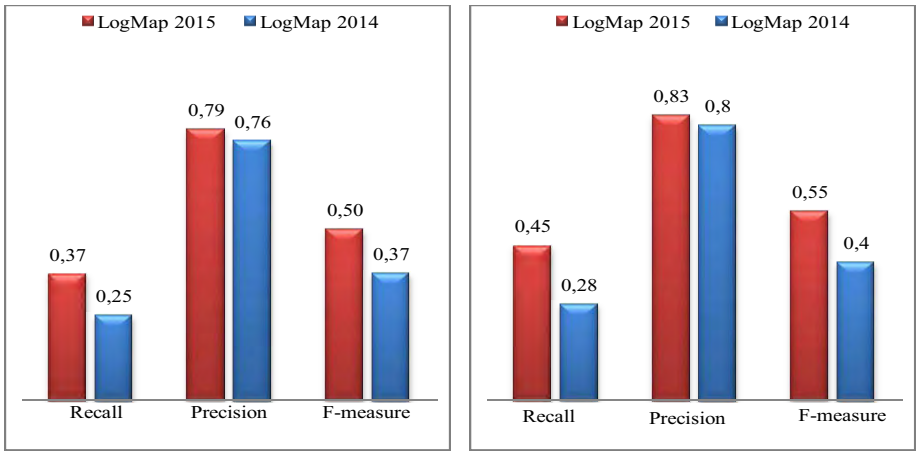
3.3 Step 3: Validation by a Linguistic Expert

Our dataset was validated by a linguistic expert with regard to the translation of concepts and properties. Furthermore we also checked the correctness of the new reference alignments.

[2] Note that, in the literature one can find datasets in Arabic language applied to other domains different from Ontology Matching (e.g. [15, 16])

4 Experimental Study

In order to evaluate the ADOM dataset, we have used the LogMap system which is one of the top ontology alignment systems on multifarm track (see Table 1). The purpose of this evaluation is to show that the ADOM dataset is suitable to test ontology matching systems that implement multilingual support.



(a) LogMap results on ADOM dataset (b) LogMap System on Multifarm Track

Fig. 1: LogMap 2014 and 2015 results on ADOM and in all multifarm dataset.

We have tested the ADOM dataset with two versions of LogMap system. The first version, which has participated in the OAEI 2014, uses the Google translator API. The second, which aims at participating in the OAEI 2015, uses both the Microsoft and Google translators APIs. Figure 1 summarizes the average results, in terms of precision, recall and F-measure, obtained by LogMap on the ADOM dataset (Fig. 1a) and on all multifarm tests (Fig. 1b). We can appreciate that, on average, the ADOM dataset brings an additional complexity to the multifarm track, with regard the results obtained by LogMap. Note that, we aim at obtaining a more comprehensive evaluation during the OAEI 2015 evaluation campaign to confirm this fact.

5 Conclusion

In this paper we have presented ADOM, a new dataset in Arabic language describing the conference domain. This dataset has been created for two purposes 1) studying and developing specific ontology alignment methods to align ontologies in Arabic language, 2) evaluating the ability of state of the art ontology matching systems to deal with ontologies in Arabic. The experimental study shows that ADOM dataset is suitable in practice. Furthermore, ADOM has already been integrated within the multifarm dataset and it will be evaluated in the OAEI 2015 campaign.

As future challenges, we aim at (1) developing a large corpus of ontologies and dictionaries for the Arabic language, (2) adapting state of the art NLP tools to align ontologies in Arabic language, (3) improving the state of the art translators dedicated to the Arabic language.

References

1. J. Euzenat and P. Shvaiko, “Ontology Matching”, Springer-Verlag, Heidelberg, 2013.
2. W. Djeddi and M. T.Khadir, “XMap++ results for OAEI 2014”. In Proceedings of the 9th International Workshop on Ontology Matching ISWC 2014, pp. 163169, Italy, 2014.
3. E. Jiménez-Ruiz, B. C. Grau, W Xia, A. Solimando, X. Chen, V. Cross, Y. Gong, S. Zhang and A. Chennai-Thiagarajan, “LogMap family results for OAEI 2014”. In Proceedings of the 9th Workshop on Ontology Matching ISWC 2014, pp. 126-134, Italy, 2014.
4. D. Faria, C. Martins, A. Nanavaty, A. Taheri, C. Pesquita, E. Santos, I. F. Cruz and F. M. Couto, “AgreementMakerLight results for OAEI 2014”. In Proceedings of the 9th Workshop on Ontology Matching ISWC 2014, pp. 105-112, Italy, 2014.
5. T. Buckwalter, “Arabic Morphological Analyzer Version 2.0”. LDC catalog number LDC2004L02, 2004.
6. A. Farghaly, “Arabic NLP: Overview, state of the art, challenges and opportunities”, In The International Arab Conference on Information Technology, ACIT2008, Tunisia, 2008.
7. D. Ngo and Z. Bellahsene, “YAM++ results for OAEI 2013”, In Proceedings of the 8th Workshop on Ontology Matching ISWC 2013, pp. 211-218, Australia, 2013.
8. P. Bouquet, J. Euzenat, E. Franconi, L. Serafini, G. Stamou and S. Tessaris “Specification of a Common Framework for Characterizing Alignment”, Deliverable 2.2.1, Knowledge Web NoE, Technical Report, Italy, 2004.
9. E. Jiménez-Ruiz et al. “LogMap family results for OAEI 2015”. In Proceedings of the 10th Workshop on Ontology Matching ISWC 2015, pp., USA, 2015.
10. E. Jiménez-Ruiz, Bernardo Cuenca Grau, Yujiao Zhou and Ian Horrocks. “Large-Scale Interactive Ontology Matching: Algorithms and Implementation”. In: ECAI. 2012.
11. Z. Dragisic, K. Eckert, J. Euzenat, D. Faria, A. Ferrara, R. Granada, V. Ivanova, E. Jiménez-Ruiz, A. O. Kempf, P. Lambrix, S. Montanelli, H. Paulheim, D. Ritze, P. Shvaiko, A. Solimando, C. Trojahn-dos-Santos, O. Zamazal and B. Cuenca Grau, “Results of the Ontology Alignment Evaluation Initiative 2014”, 9th Workshop on Ontology Matching, 2014.
12. B. Cuenca Grau, Z. Dragisic, K. Eckert, J. Euzenat, A. Ferrara, R. Granada, V. Ivanova, E. Jiménez-Ruiz, A. Oskar Kempf, P. Lambrix, A. Nikolov, H. Paulheim, D. Ritze, F. Scharffe, P. Shvaiko, C. Trojahn dos Santos, O. Zamazal, “Results of the Ontology Alignment Evaluation Initiative 2013”. 8th Workshop on Ontology Matching, 2013.
13. O. Svab, V. Svatek, P. Berka, D. Rak and P. Tomasek, “OntoFarm: Towards an Experimental Collection of Parallel Ontologies”, In: Poster Track of ISWC 2005, Galway, 2005.
14. C. Meilicke, R. Garca-Castro, F. Freitas, WR. Van Hage, E. Montiel-Ponsoda, R.R. De Azevedo, H. Stuckenschmidt, O. vb-Zamazal, V. Svtek and A. Tamilin, “MultiFarm: A benchmark for multilingual ontology matching”. Web Semant. Sci. Serv. Agents World Wide Web. Vol. 15, pp. 6268, 2012.
15. I. Bounhas, B. Elayeb, F. Evrard, and Y. Slimani, ”ArabOnto: Experimenting a new distributional approach for Building Arabic Ontological Resources”. In International Journal of Metadata, Semantics and Ontologies, Inder-science, Vol. 6, No. 2, pp. 81-95, 2011.
16. O. Ben Khiroun, R. Ayed, B. Elayeb, I. Bounhas, N. Bellamine Ben Saoud and F. Evrard, “Towards a New Standard Arabic Test Collection for Mono- and Cross-Language Information Retrieval”, In the Proceedings of 19th International Conference on Application of Natural Language to Information Systems (NLDB), 2014.

Ontology Matching for Big Data Applications in the Smart Dairy Farming Domain

Jack P.C. Verhoosel, Michael van Bekkum and Frits K. van Evert

TNO Connected Business, Soesterberg, The Netherlands
{jack.verhoosel,michael.vanbekkum}@tno.nl
Wageningen UR, Wageningen, The Netherlands
frits.vanevert@wur.nl

Abstract. This paper addresses the use of ontologies for combining different sensor data sources to enable big data analysis in the dairy farming domain. We have made existing data sources accessible via linked data RDF mechanisms using OWL ontologies on Virtuoso and D2RQ triple stores. In addition, we have created a common ontology for the domain and mapped it to the existing ontologies of the different data sources. Furthermore, we verified this mapping using the ontology matching tools HerTUDA, AML, LogMap and YAM++. Finally, we have enabled the querying of the combined set of data sources using SPARQL on the common ontology.

1 Background and context

Dairy farmers are currently in an era of precision livestock farming in which information provisioning for decision support is becoming crucial to maintain a competitive advantage. Therefore, getting access to a variety of data sources on and off the farm that contain static and dynamic individual cow data is necessary in order to provide improved answers on daily questions around feeding, insemination, calving and milk production processes.

In our SmartDairyFarming project, we have installed sensor equipment to monitor around 300 cows each at 7 dairy farms in The Netherlands. These cows have been monitored during the year 2014 which has generated a huge amount of sensor data on grazing activity, feed intake, weight, temperature and milk production of individual cows stored in databases at each of the dairy farms. The amount of data recorded per cow is at least 1MB of sensor values per month, which adds up to 3.6GB of data per dairy farm per year. In addition, static cow data is available in a data warehouse at the national milk registration organization, including date of birth, ancestors and current farm. Finally, another existing data source contains satellite information on the amount of biomass in grasslands in the country that is important for measuring the feed intake of cows during grazing.

We focused on decision support for the dairy farmer on feed efficiency in relation to milk production. Thus, the big data analysis question is: “How much feed did an individual cow consume in a certain time period at a specific grassland parcel and how does this relate to the milk production in that period?”.

2 Ontology matching approach

We selected one of the dairy farms (DairyCampus) and created with TopBraid composer a small ontology with 12 concepts that covers among others the grasslands

of a farm and grazing periods of cows. This ontology contains the concept “perceel” which is Dutch for parcel. In addition, we selected the data source with satellite information about biomass in grasslands (AkkerWeb, www.akkerweb.nl). This data source already had an ontology defined with 15 concepts that contains the concept “plot” which is similar to parcel but with different properties. Furthermore, we created with TopBraid composer a common ontology for the domain with 28 concepts on feed efficiency (see **Fig. 1**).

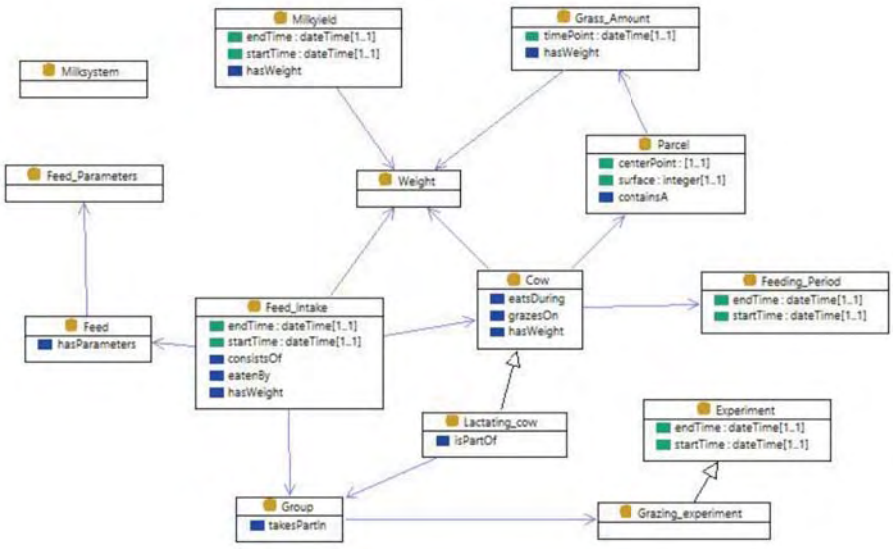


Fig. 1. Common ontology excerpt for feed efficiency in dairy farming.

The challenge was to find a match between the concepts and properties in the common ontology and both specific DairyCampus and Akkerweb ontologies, especially regarding the concepts “parcel”, “perceel” and “plot”.

We have initially created manual mappings between classes and properties in TopBraid using rdfs:subClassOf and owl:equivalentProperty relations. Based on relatively few and simple matches we created initial alignments between properties and classes (see **Fig. 2**).

Use of a matching tool or system however, provides us with opportunities to verify our current findings and better support our efforts in finding alignments between the other concepts in our ontologies. We used a literature survey of matching techniques and supporting matching systems in [1] to identify both a suitable matching technique and find tools supporting that technique. We consider language-based matching as the appropriate type of matching since it focuses on syntactic element-level natural language processing of words.

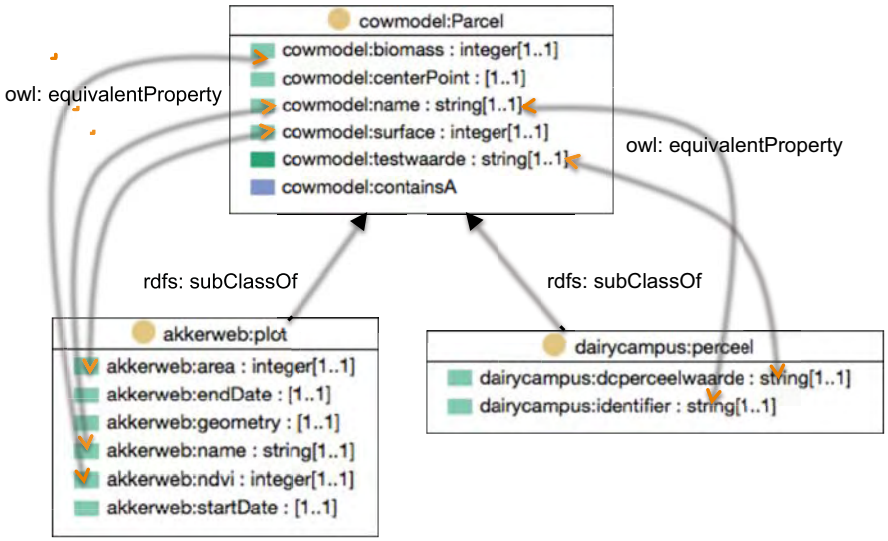


Fig. 2. Mapping of classes and properties based on the matching result.

There are numerous tools available that support this specific matching technology, mostly from academic efforts. Some however are no longer in active use, either being outdated or not maintained anymore [2].

We have selected several matching systems that support our requirement of language-based matching: HerTUDA [3,4], AgreementMaker Light (AML) [5], LogMap [6], and YAM++ [7]. We have started to investigate the possibilities of these tools to find alignments of concepts and properties in our ontologies. Initial efforts with the concepts shown in **Fig. 2** have not led to successful matches and alignments yet, however. The HerTUDA, LogMap and YAM++ tools were difficult to install and execute. The AML worked fine, but could not entirely find the relation between “parcel”, “perceel” and “plot”. Further analysis is required to find out whether this is due to inappropriate matching techniques or to the specific ontologies that we offered to the tool.

3 ! SPARQL queries and triple stores

In order to show that the mapping of the common ontology to the specific ontologies works properly, we generated in TopBraid a few instances of an Akkerweb plot and a DairyCampus perceel. In addition, we build a simple select query using the common ontology to retrieve all parcels and for each parcel the properties name, biomass, surface and test.

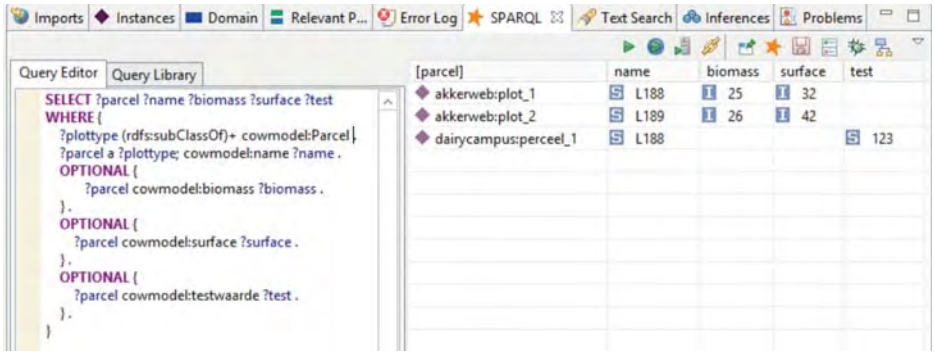


Fig. 3. Select query on common ontology to retrieve all parcels.

The query and its results are shown in **Fig. 3**. As can be seen, the query retrieves both Akkerweb plots and DairyCampus percelen. In addition, Akkerweb contains data about a plot with name "L188" and DairyCampus contains data on a perceel with an identifier "L188". This means that both databases contain the same parcel and the properties can be combined.

The specific ontologies for DairyCampus and Akkerweb formed the basis to generate triples from the relational data sources of DairyCampus and Akkerweb. The triples have been made available via Virtuoso as well as directly from the D2RQ tool (www.d2rq.org). A system that is based on the common ontology can take the big data question to create federated SPARQL queries on the DairyCampus and Akkerweb triple stores using the matched ontologies. As a result, farmers can pose questions in terms of the concepts in the common ontology instead of the detailed and specific concepts of the DairyCampus and Akkerweb data sources.

The farmer can use such a system for decision support purposes on various daily operations, such as which amount of feed to provide to which cow in which period, when to inseminate a specific cow and how to deal with the transition of a cow towards calving.

4 Future work

The approach that is describe in this paper is currently in an experimental phase. We have reached a set-up by filling the triple stores for 3 farms with cow-data of 1 month which adds up to a total of 7 million triples. This needs to be upgraded to all farms with all data from 2014. Thereby, we can test the scalability of our system. In addition, we need to do more detailed analysis of the matching tools that we used and the reasons for not adequately solving the simple matching problem that we proposed.

References

1. ! Otero-Cerdeira, L., Rodriguez-Martinez, F.J., Gomez-Rodriguez, A.: Ontology matching: A literature review. Journal on Expert Systems with Applications, 949-971 (2015)
2. ! Ontology matchings tool overview: www.mkbergman.com/1769/50-ontology-mapping-and-alignment-tools/

3. ! Hertling, S.: Hertuda results for OAEI 2012. In Ontology Matching 2012 workshop proceedings, 141-144 (2012)
4. ! HerTUDA download: www.ke.tu-darmstadt.de/resources/ontology-matching/hertuda
5. ! AgreementMakerLight website: somer.fc.ul.pt/aml.php
6. ! LogMap website: www.cs.ox.ac.uk/isg/tools/LogMap/
7. ! YAM++ website: www.lirmm.fr/yam-plus-plus

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

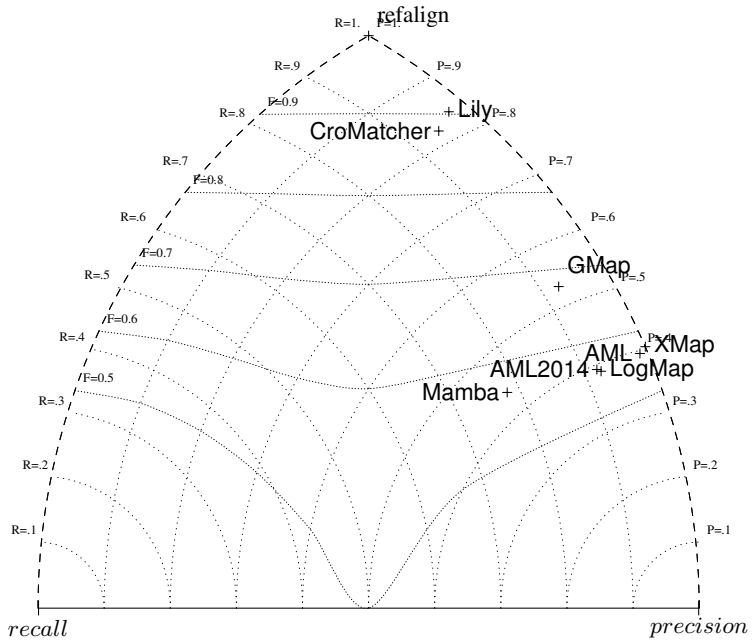


Fig. 1. Triangle view on the benchmark biblio data sets (run 5, non present systems have too low F-measure).

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

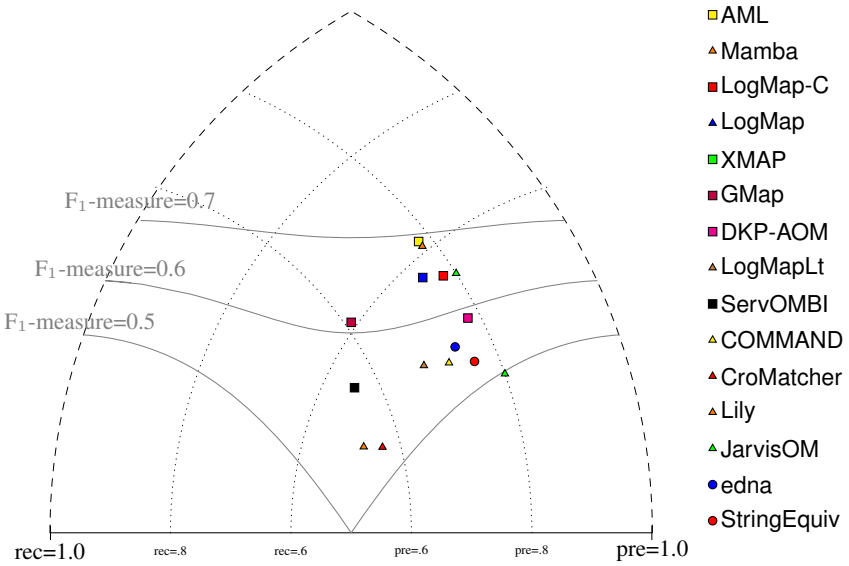


Fig. 2. Precision/recall triangular graph for the conference test case wrt. the *rar2* reference alignment. Dotted lines depict level of precision/recall while values of F_1-measure are depicted by areas bordered by corresponding lines F_1-measure=0.[5|6|7].

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 advantage* 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

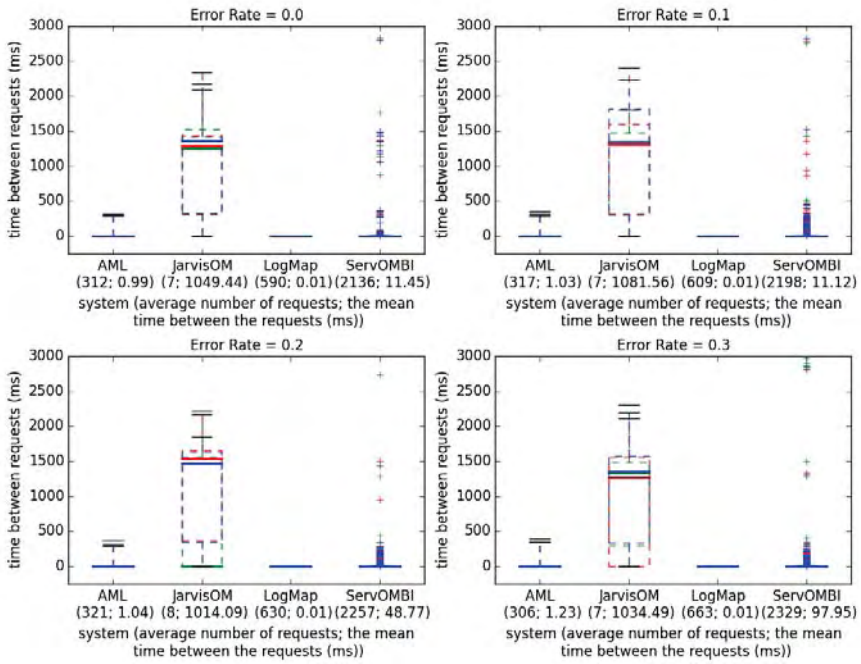


Fig. 3. The Y axis depicts the time intervals between the requests to the user/oracle (whiskers: Q1-1,5IQR, Q3+1,5IQR, IQR=Q3-Q1). The labels under the system names show the average number of requests and the mean time between the requests for the three runs.

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

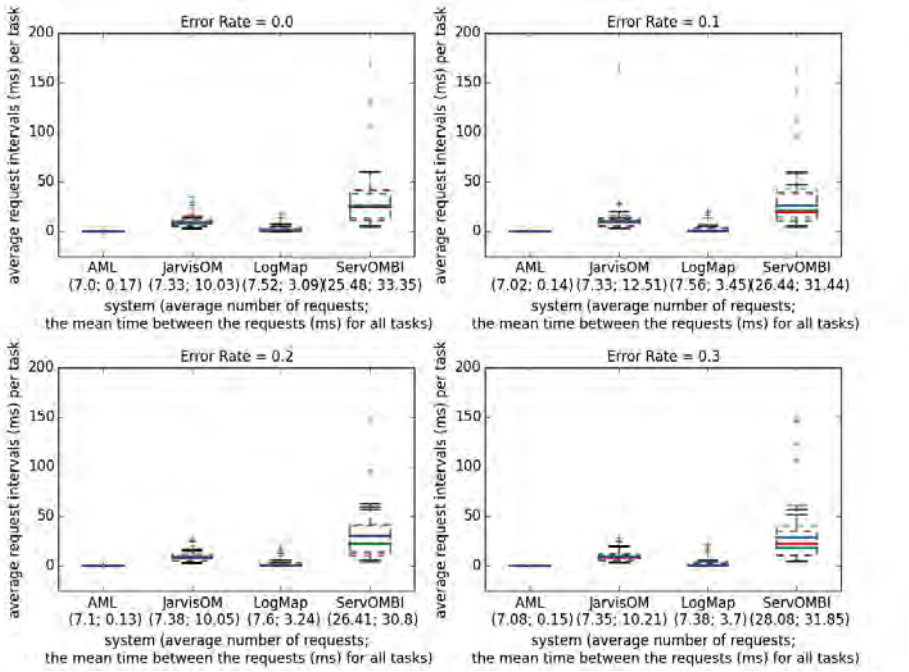


Fig. 4. The Y axis depicts the average time between the requests per task in the Conference dataset (whiskers: Q1-1,5IQR, Q3+1,5IQR, IQR=Q3-Q1). The labels under the system names show the average number of requests and the mean time between the requests (calculated by taking the average of the average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

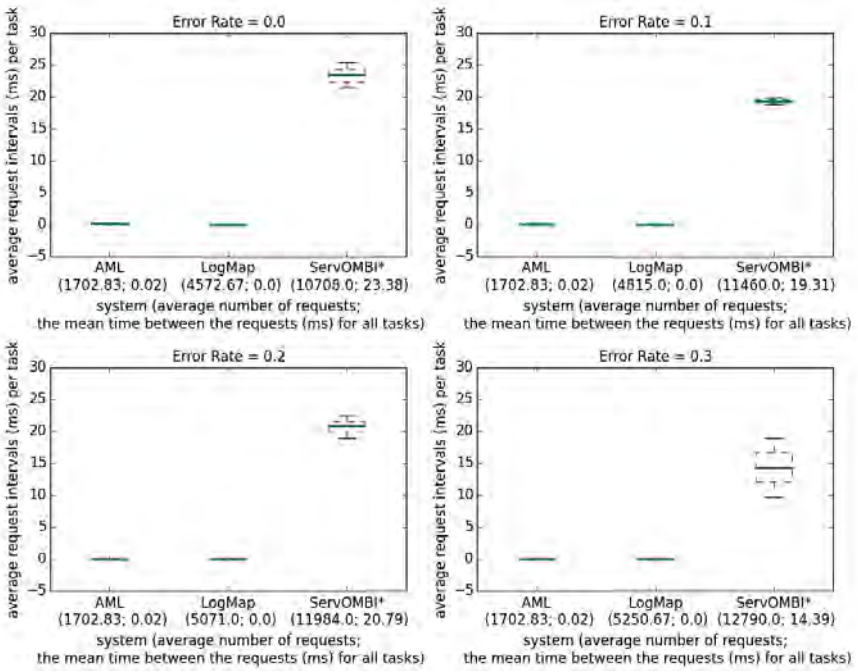


Fig. 5. The Y axis depicts the average time between the requests per task in the largebio dataset (6 tasks) (whiskers: Q1-1,5IQR, Q3+1,5IQR, IQR=Q3-Q1). The labels under the system names show the average number of requests and the mean time between the requests (calculated by taking the average of the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

AML Results for OAEI 2015

Daniel Faria[1], Catarina Martins[2], Amruta Nanavaty[3]
, Daniela Oliveira[2], Booma S. Balasubramani[3], Aynaz Taheri[3]
, Catia Pesquita[2], Francisco M. Couto[2], and Isabel F. Cruz[3]

[1] Instituto Gulbenkian de Ciência, Portugal
[2] LaSIGE, Faculdade de Ciências, Universidade de Lisboa, Portugal
[3] ADVIS Lab, Department of Computer Science, University of Illinois at Chicago, USA

Abstract. AgreementMakerLight (AML) is an automated ontology matching system based primarily on element-level matching and on the use of external resources as background knowledge. This paper describes its configuration for the OAEI 2015 competition and discusses its results.
For this OAEI edition, we focused mainly on the Interactive Matching track due to its expansion, as handling user interactions on large-scale tasks is a critical challenge in ontology matching.
AML's participation in the OAEI 2015 was successful, as it obtained the highest F-measure in 6 of the 7 ontology matching tracks. Notably, it obtained the highest F-measure in all tasks of the Interactive Matching track while posing less queries to the user than comparable participating systems.

1 Presentation of the system

1.1 State, purpose, general statement

AgreementMakerLight (AML) is an automated ontology matching system based primarily on lexical matching techniques, with an emphasis on the use of external resources as background knowledge and on alignment coherence. While originally focused on the biomedical domain, AML's scope has been expanded, and it can now be considered a general-purpose ontology matching system, as evidenced by its results in last year's OAEI.
AML was derived from AgreementMaker [1, 2] and combines its design principles (flexibility and extensibility) with a strong focus on efficiency and scalability [5]. It draws on the knowledge accumulated in AgreementMaker by reusing and adapting some of its components, but also includes a number of novel components such as an alignment repair module [11] and an automatic background knowledge source selection algorithm [4].
This year, our development of AML for the OAEI competition focused primarily on the Interactive Matching track, due to its expansion to include the Anatomy and Large Biomedical Ontologies datasets. Handling user feedback on large-scale tasks is a critical challenge in ontology matching, and was an aspect in which AML still had room for improvement.

1.2 Specific techniques used

The AML workflow for the OAEI 2015 is the same as last year, comprising the nine steps shown in Figure 1: ontology loading and profiling, translation, baseline matching, background knowledge matching, word and string matching, structural matching , property matching, selection, and repair.

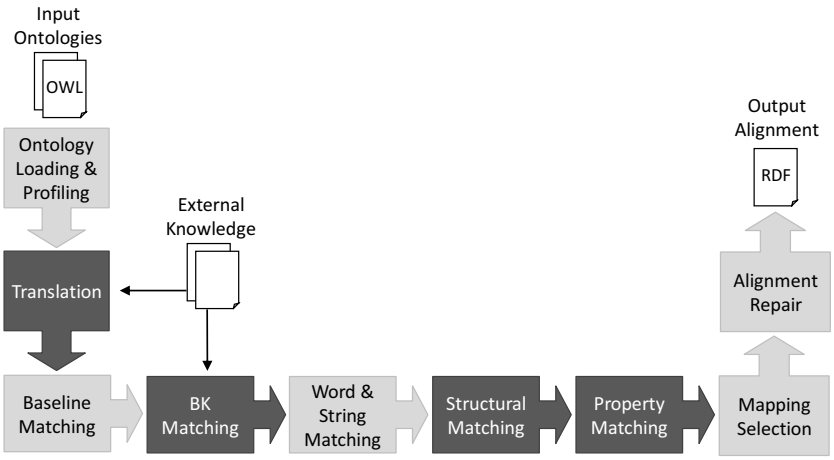


Fig. 1. The AgreementMakerLight matching workflow for the OAEI 2015. Steps in dark gray are conditional.

Ontology Loading & Profiling AML employs the OWL API [6] to read the input ontologies and retrieve the necessary information to populate its own data structures [5]:

- Local names, labels and synonym annotations of Classes, Object Properties and Data Properties are normalized and stored into the *Lexicon* of the corresponding ontology. AML automatically derives new synonyms for each name by removing leading and trailing stop words [10], and by removing name sections within parenthesis.
- Domains and ranges of Object and Data Properties are stored in the *Ontology* in *Property* objects.
- Relations between classes (including disjointness) and between properties are stored in a global *RelationshipMap*.
- Cases of implicit disjointness between classes that have incompatible property restrictions in their definition (e.g., different values of a Functional Data Property such as *has mass*) are inferred and made explicit in the *RelationshipMap* as well.

AML does not store or use comments, definitions, or instances.
After loading, the matching problem is profiled taking into account the size of the ontologies, their language(s), and their property/class ratio.

Translation AML features an automatic translation module based on Microsoft® Translator, which is called when there is no significant overlap between the language(s) of the input ontologies. AML employs this module to translate the names of all classes and properties from the language(s) of the first ontology to the language(s) of the second and vice-versa. The translation is done by querying Microsoft® Translator for the full name (rather than word-by-word) in order to help provide context. To improve performance, AML employs a cache strategy, by storing locally all translation results in dictionary files, and queries the Translator only when no stored translation is found.

Baseline Matching AML employs an efficient, and generally precise, weighted string-equivalence algorithm, the *Lexical Matcher* [5], to obtain a baseline class alignment between the input ontologies.

Background Knowledge Matching AML has available four sources of background knowledge which can be used as mediators between the input ontologies: the Uber Anatomy Ontology (Uberon) [8], the Human Disease Ontology (DOID) [12], the Medical Subject Headings (MeSH) [9], and the WordNet [7].
The WordNet is only used for small English language ontologies, as it is prone to produce erroneous mappings in other settings (particularly in domains with specialized vocabularies, such as the Life Science domain). It is used through the JAWS API [1] and with the *Lexical Matcher*. The remaining three background knowledge sources are all specific to the biomedical domain, and thus are tested for all non-small English language ontologies, given that biomedical ontologies are seldom small. They are tested by measuring their mapping gain over the baseline alignment [4]. When the mapping gain is high ($\geq$20%), the source is used to extend the *Lexicons* of the input ontologies [10]; otherwise, when it is above the minimum threshold (2%) they are used merely as mediators and their alignment is added to the baseline alignment.
Uberon and DOID are both used in OWL format, and each has an additional table of pre-processed cross-references (in a text file). They can be used directly through the cross-references or with the *Lexical Matcher*. MeSH is used as a stored *Lexicon* file, which was produced by parsing its XML file, and is used only with the *Lexical Matcher*.

Word & String Matching To further extend the alignment, AML employs a word-based similarity algorithm (the *Word Matcher*) and a string similarity algorithm (the *Parametric String Matcher*) [5]. The former is not used for very large ontologies, because it is error prone. The latter is used globally for small ontologies, but only locally for larger ones as it is time-consuming.
For small ontologies, AML also employs the *Multi-Word Matcher*, which matches closely related multi-word names that have matching words and/or words with common WordNet synonyms or close hypernyms, and the new *Acronym Matcher*, which attempts to match acronyms to the corresponding full name.

[1] http://lyle.smu.edu/ tspell/jaws/

Structural Matching For small and medium-sized ontologies, AML also employs a structural matching algorithm, called *Neighbor Similarity Matcher*, that is analogous to AgreementMaker's Descendants Similarity Inheritance algorithm [3]. This algorithm computes similarity between two classes by propagating the similarity of their matched ancestors and descendants, using a weighting factor to account for distance.

Property Matching When the input ontologies have a high property/class ratio, AML also employs the *PropertyMatcher*. This algorithm first ensures that properties have the same type and corresponding/matching domains and ranges. If they do, it compares the properties' names by doing a full-name match and computing word similarity, string similarity, and WordNet similarity.

Selection AML employs a greedy selection algorithm, the *Ranked Selector* [5], to reduce the cardinality of the alignment. Depending on the size of the input ontologies, one of three selection strategies is used: strict, permissive, or hybrid. In strict selection, no concurrent mappings (i.e., different mappings for the same class/property) are allowed and a strict 1-to-1 alignment is produced; in permissive selection, concurrent mappings are allowed if their similarity score is exactly the same; in hybrid selection, up to two mappings per class are allowed above 75% similarity, and permissive selection is applied below this threshold. For very large ontologies, AML employs a selection variant that consists on combining the (lexical) similarity between the classes with their structural similarity, prior to performing ranked selection. This strategy enables AML to select mappings that "fit in" structurally over those that are outliers but have a high lexical similarity.
In interactive matching mode, AML employs an interactive selection algorithm instead. This algorithm uses patterns in the similarity values produced by AML's various matching algorithms to detect suspicious mappings. Above the high similarity threshold of 70%, AML queries the user for suspicious mappings, and accepts all other mappings as true. Below this threshold, AML automatically rejects suspicious mappings, and queries the user for all other mappings, until the minimum threshold of 45% is reached, the limit of consecutive negative answers is reached, or the query limit is reached, whichever happens first. The query limit is 45% of the alignment for small ontologies, and 15% of the alignment for all other ontologies (with a further 5% of the alignment reserved for interactive repair). It ensures that the workload for the user is kept within reasonable boundaries.

Repair AML employs a heuristic repair algorithm to ensure that the final alignment is coherent [11].
For the interactive matching track, AML employs an interactive variant of this algorithm, wherein the user is asked for feedback about the mappings selected for removal. This variant is not used on the Large Biomedical Ontologies dataset due to its particular evaluation, wherein mappings repaired from the reference alignment are ignored but considered true by the Oracle.

1.3 Adaptations made for the evaluation

The only adaptations made for the evaluation were the preprocessing of cross-references from Uberon and DOID for use in the Anatomy and Large Biomedical Ontologies tracks (due to namespace differences), and the precomputing of translations for the Multifarm track (due to Microsoft® Translator's query limit).

1.4 Link to the system and parameters file

AML is an open source ontology matching system and is available through GitHub (https://github.com/AgreementMakerLight) as an Eclipse project, as a stand-alone Jar application, and as a package for running through the SEALS client.

2 Results

2.1 Anatomy

AML had almost identical results to last year, with an F-measure of 94% and a recall++ of 82%, making it the best performing system in this track this year as well. The only difference from last year's alignment was one missing mapping due to a change in the structural matching algorithm.

2.2 Benchmark

AML had a small improvement in the Biblio Benchmark over last year, from 55% to 57%, likely due to the few refinements made in the processing of properties. However, its performance on the new Energy Benchmark was poor, with a recall of only 2%, and consequently a low F-measure as well (18%). This remains the only OAEI track where AML's performance is sub-par, mainly due to the fact that involves instances, which AML currently does not read or process in any way.

2.3 Conference

AML had the best performance overall in the Conference track, with the highest F-measure on the full reference alignments ra1 and ra2 (74% and 70% respectively). It also had the highest F-measure in the class-only alignments, and the second-highest in the property-only alignments (notably with 100% precision). In comparison with last year, AML improved its F-measure by 3% with regard to ra2, thanks to the addition of the *Acronym Matcher* and to a few refinements in the processing of properties. Concerning the logical reasoning evaluation, AML was one of the five systems that produced alignments without consistency principle violations, and it had an average number of conservativity principle violations of 1.86 which is the sixth lowest overall, and a reasonable figure considering that some of these violations are false positives.

2.4 Interactive Matching

AML obtained the highest F-measure in all interactive tasks, with 96.2% in Anatomy (with no error), 81.8% in Conference and an average of 84.5% in LargeBio. AML also had the lowest number of queries among comparable systems in all datasets (i.e., LogMap and ServOMBI, as JarvisOM called upon the Oracle in an active learning approach rather than to filter mapping candidates, which enabled it to make a minimal number of queries in Anatomy, but resulted in it having the worst F-measure as well). It should be noted, however, that AML had the highest non-interactive F-measure on all tracks, so it is unsurprising that it could remain ahead of the other systems while making less queries. Thus, it is important to add that AML also had the highest F-measure-gain-per-query ratio among comparable systems in all datasets (again, excluding JarvisOM), meaning it was more efficient in exploring the user feedback.
With regard to the introduction of Oracle errors, AML was the only system where their impact was linear, with all other systems being impacted superlinearly. The evidence lies in the fact that AML's F-measure was approximately constant when evaluated by the Oracle (i.e., when considering the errors made by the Oracle to be correct) whereas the other systems' F-measures decreased as the error increased. This implies that other systems are drawing inferences from the Oracle's replies, and deciding on the outcome of multiple mappings based on a single query, whereas AML is treating each mapping more or less independently, and thus is less sensitive to the impact of Oracle errors.

2.5 Large Biomedical Ontologies

AML's performance in this track was exactly the same as last year, with an average F-measure of 81.9%, as none of the developments made affect this track. As last year, AML had the highest F-measure in each individual task (among valid participants), and thus the highest average F-measure as well. Furthermore, it also had the lowest average degree of unsatisfiabilities, though it was closely followed by LogMap.

2.6 Multifarm

AML had an F-measure of 51% when matching different ontologies and of 64% when matching the same ontologies in different languages, both of which were the highest overall by a considerable margin (the next best system in matching different ontologies was LogMap at 41% F-measure, and at matching the same ontologies was CLONA at 58% F-measure). It also had the highest recall overall in both modes, and the second-highest precision. These results are not directly comparable to last year, due to the introduction of the Arabic language ontologies, but running this year's AML on last year's dataset, we observe a marginal improvement in matching different ontologies (by 0.1% F-measure) but a substantial improvement in matching the same ontologies (by 3.3% F-measure). This improvement is mainly due to the refinements made to structural matching algorithm, which naturally have a higher impact on matching different languages of the same ontology, given that the structure will be the same.

2.7 Ontology Alignment for Query Answering

AML had the best performance in this track this year, with an F-measure of 75.9% using the original reference alignment (ra1) and 74.4% using the repaired reference alignment (rar1). It also had the highest precision (tied with XMap on ra1) and recall (tied with LogMap on both ra1 and rar1). These results reflect the fact that AML was the best performing system in the Conference track, and therefore, is naturally the system best positioned to use its Conference alignments for query answering.

3 General comments

3.1 Comments on the results

In comparison with last year, AML improved its performance in 5 tracks: Benchmark (Biblio dataset), Conference, Interactive Matching, Multifarm, and Ontology Alignment for Query Answering. It's performance in the Anatomy and LargeBio tracks was essentially the same as last year. These improvements are tied to developments made in structural matching, property processing and matching, and interactive selection, which reflect the effort put into AML for this year's OAEI.

3.2 Discussions on the way to improve the proposed system

While AML has established itself as a versatile and effective ontology matching system, there is still an important aspect where it is lacking: handling and matching ontology instances.

3.3 Comments on the OAEI test cases

The expansion of the Interactive Matching track to include more challenging test cases and simulate user error was an important improvement to this track and to the OAEI as a whole. Alas, not all was perfect with this year's evaluation, as the Oracle's behaviour on the LargeBio 'soft' repaired reference alignments severely hindered the performance of any interactive repair algorithm, and led to our decision not to employ ours on the LargeBio datasets. We also believe that a query limit should be enforced to ensure that the usage of the Oracle remains within reasonable boundaries, so that systems cannot employ the Oracle to review all their mapping candidates.

4 Conclusion

For this OAEI edition, our goal was to improving AML's interactive selection algorithm and refine its strategy for matching small ontologies. We decided not to make any developments for the biomedical tracks (Anatomy and Large Biomedical Ontologies) as AML's performance was already very good, and we felt that investing further in these tracks would bring a low return on investment.
The results obtained by AML this year have reflected and rewarded our effort, topping

the tables with regard to F-measure in all ontology matching tasks except for Benchmark, with improvements upon last year's performance in the Interactive Matching track and all tracks based on the Conference dataset, while maintaining the performance in Anatomy and Large Biomedical Ontologies.
Thus the OAEI 2015 results highlight the fact that AML is an effective, efficient, and versatile ontology matching system.

Acknowledgments

FMC, CM, DO and CP were funded by the Portuguese FCT through the LASIGE Strategic Project (UID/CEC/00408/2013). The research of IFC, AN, BS and AT was partially supported by NSF Awards CCF-1331800, IIS-1213013, and IIS-1143926.

References

1. I. F. Cruz, F. Palandri Antonelli, and C. Stroe. AgreementMaker: Efficient Matching for Large Real-World Schemas and Ontologies. *PVLDB*, 2(2):1586–1589, 2009.
2. I. F. Cruz, C. Stroe, F. Caimi, A. Fabiani, C. Pesquita, F. M. Couto, and M. Palmonari. Using AgreementMaker to Align Ontologies for OAEI 2011. In *ISWC International Workshop on Ontology Matching (OM)*, volume 814 of *CEUR Workshop Proceedings*, pages 114–121, 2011.
3. I. F. Cruz and W. Sunna. Structural alignment methods with applications to geospatial ontologies. *Transactions in GIS*, 12(6):683–711, 2008.
4. D. Faria, C. Pesquita, E. Santos, I. F. Cruz, and F. M. Couto. Automatic Background Knowledge Selection for Matching Biomedical Ontologies. *PLoS One*, 9(11):e111226, 2014.
5. D. Faria, C. Pesquita, E. Santos, M. Palmonari, I. F. Cruz, and F. M. Couto. The AgreementMakerLight Ontology Matching System. In *OTM Conferences - ODBASE*, pages 527–541, 2013.
6. M. Horridge and S. Bechhofer. The owl api: A java api for owl ontologies. *Semantic Web*, 2(1):11–21, 2011.
7. G. A. Miller. WordNet: A Lexical Database for English. *Communications of the ACM*, 38(11):39–41, 1995.
8. C. J. Mungall, C. Torniai, G. V. Gkoutos, S. Lewis, and M. A. Haendel. Uberon, an Integrative Multi-species Anatomy Ontology. *Genome Biology*, 13(1):R5, 2012.
9. S. J. Nelson, W. D. Johnston, and B. L. Humphreys. Relationships in medical subject headings (mesh). In *Relationships in the organization of knowledge*, pages 171–184. Springer, 2001.
10. C. Pesquita, D. Faria, C. Stroe, E. Santos, I. F. Cruz, and F. M. Couto. What's in a "nym"? Synonyms in Biomedical Ontology Matching. In *International Semantic Web Conference (ISWC)*, pages 526–541, 2013.
11. E. Santos, D. Faria, C. Pesquita, and F. M. Couto. Ontology alignment repair through modularization and confidence-based heuristics. arXiv:1307.5322, 2013.
12. L. M. Schriml, C. Arze, S. Nadendla, Y.-W. W. Chang, M. Mazaitis, V. Felix, G. Feng, and W. A. Kibbe. Disease Ontology: a backbone for disease semantic integration. *Nucleic Acids Research*, 40(D1):D940–D946, 2012.

Clona Results for OAEI 2015

Mariem El Abdi, Hazem Souid, Marouen Kachroudi
and Sadok Ben Yahia

Université de Tunis El Manar, Faculté des Sciences de Tunis, LIPAH Programmation Algorithmique et Heuristique, 2092, Tunis, Tunisie;
elabdi.mariam@gmail.com
swdhazem@gmail.com
{marouen.kachroudi,sadok.benyahia}@fst.rnu.tn

Abstract. This paper presents the results of Clona in the Ontology Alignment Evaluation Initiative campaign (OAEI) 2015. We only participated in Multifarm track, since Clona develops specific techniques for aligning multilingual ontologies. We first give an overview of our alignment system; then we detail the techniques used in our contribution to deal with cross-lingual ontology alignment. Last, we present the results with a thorough analysis and discussion, then we conclude by listing some future work on Clona.

1 Presentation of the system

Multilingualism has become an issue of major interest for the Semantic Web community. This process has been accelerated due to a few initiatives which encourage all the active participants to make their data available to the public. Multilingualism is identified as one of the six challenges of the Semantic Web. Consequently, some solutions were proposed at the ontology level, annotation level and the interface level [1].

At the ontology level, the support should be conceived by the ontology designers to create knowledge representations in diverse natural languages. At the annotation level, tools should be developed to assist users in ontologies annotating independently of the natural languages adopted in their design and development. At the interface level, users should be able to have access to the information in natural languages of their own choice, without any linguistic restriction. The absence of the multilingual aspect coverage can be a real handicap during the information exchange in between various services offered by the Semantic Web [2]. So, application fields are more and more numerous and they put in front very specific difficulties. Moreover, the multilingualism coverage allows the reasoning on the context intersections of various ontological representations. In this register, the issue of reasoning on overlapping context domains led to support multilingual information retrieval and digital content management. Multilingual

ontologies alignment is still a little investigated domain in spite of the multiplicity of the alignment methods which remain restricted to monolingual ontologies [3–6].

CLONA as a few methods [7–10] meets challenges strictly bound at the linguistic level in the context of multilingual ontology alignment. The driven idea of our new method is to cross the natural language barrier. CLONA presents a novel view to improve the alignment accuracy that draws on the information retrieval techniques.

1.1 State, purpose, general statement

The CLONA workflow for the OAEI 2015 comprises six different steps, as flagged by Figure 1 : (**i**) Parsing and Pretreatment, (**ii**) Translation, (**iii**) Indexation, (**iv**) Candidate Mappings Identification and (**vi**) Alignment Generation.

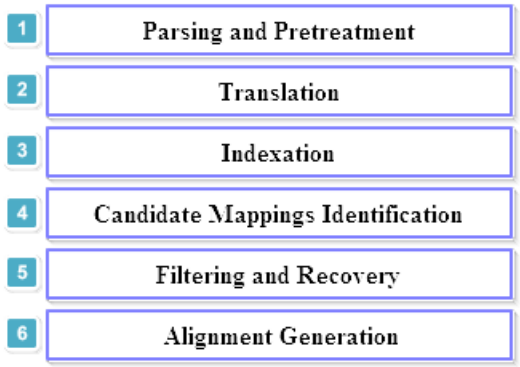


Fig. 1. CLONA workflow for OAEI 2015 (Multifarm Track)

CLONA is an alignment system which aims through specific techniques to identify the correspondences between two ontologies defined in two different natural languages. Indeed, it starts with a pretreatment stage to model the input ontologies by a format for the rest of the process. The second phase is that of translation into a chosen pivot language and provided by the Microsoft Bing [1] translator.

Thereafter, our method continues with an indexing phase over the considered ontologies. Then these indexes are asked to supply the candidate mappings list to be aligned. Before generating the alignment file, CLONA uses a filtering module for recovery and repair.

[1] http://www.microsoft.com/en-us/translator

Parsing and Pretreatment : This phase is crucial for ontologies pretreatment. It is performed using the OWL API. Indeed, it transforms the considered ontologies represented initially as two OWL files in an adequate format for the rest of the treatments. In our case, the goal is to remove all the existing information in both OWL files so that each entity is represented by all its properties. Indeed, the parsing module begins by loading two ontologies to align described in OWL.

This module allows to extract the ontological entities initially represented by a primitive form of lists. In other words, at the parsing stage, we seek primarily to transform an OWL ontology in a well defined structure that preserves and highlight all the information contained in this ontology. Furthermore, in the resulting informative format, has a considerable impact on the results of the similarity computation thereafter. Thus, we get couples formed by the name of the entity and its associated label. In the next step we add an element to such couples to process these entities regardless of their native language.

Translation : The main goal of our approach is to solve the heterogeneity problem mainly due to multilingualism. This challenge brings us to choose between two alternatives, either we consider the translation path to one of the languages according to the two input ontologies, or we consider the translation path to a chosen pivot language. At this stage, we must have a vision of foreseeable rest of our approach. Specifically, at the semantic alignment stage we use an external resource such as WordNet. The latter is a lexical database for the English language. Therefore, our choice is well taken, and we will prepare a translation of the two ontologies to the pivot language, which is English. To perform the translation phase we chose Bing Microsoft tool.

Indexation : Whether on the Internet, with many search engine or local access, we need to find documents or simply sites. Such research is valuable to browse each file and the analysis thereafter. However, the full itinerary of all documents with the terms of a given query is expensive since there are too many documents and prohibitive response times. To enable faster searching, the idea is to execute the analysis in advance and store it in an optimized format for the search. Indexing is one of the novelties of our approach. It consists in reducing the search space through the use of effective search strategy on the built indexes. In fact, we no longer need the sequential scan because with the index structure, we can directly know what document contains a particular word. To ensure this indexing phase we use the Lucene [2] tool. Lucene is a Java API that allows developers to customize and deploy their own indexing and search engine. Lucene uses a suitable technology for all applications that require text search. Indeed, at the end of the indexing process, we get four different indexes to everyone of the two input ontologies depending on the type of the detected entities (*i.e.*, concepts, data types, relationships, and instances). The documents at the indexes represent the se-

[2] https://lucene.apache.org/

mantic information about the entity. These semantic information is obtained by means of an external resource (*i.e.*, WordNet). Indeed, for each entity, CLONA keeps the entity name, the label, the label translated to English and its synonyms in English. So with Lucene, we created a set of indexes for the two ontologies, a search query is set up to return all the candidates.

Candidate Mappings Identification : `TermQuery` is the most basic query type to search through an index. It can be built using one term. In our case, `TermQuery`'s role is to find the entities in common between the indexes. Indeed, once the two indexes are set up, the querying step of the latter is activated. Thus, the query implementation satisfies the terminology search and semantic aspects at once as we are querying documents that contain a given ontological entity and its synonyms obtained via WordNet. The result of this process is a set of documents sorted by relevance according to the Lucene score assigned to each returned document. Thus, for each query, CLONA keep the first five documents returned and considers them as candidate mappings for the next phase.

Filtering and Recovery : The filtering module consists of two complementary sub modules, each one is responsible of a specific task in order to refine the set of aligned candidates. Indeed, once the list of candidates is ready, CLONA uses the first filter. Indeed, we should note that indexes querying may includes a set of redundant mappings. This filter eliminate this redundancy. Indeed, it goes through the list of candidates and for each candidate, it checks if it still exists in the list. If this is the case, it removes the redundant element. At the end of filtering phase, we have a candidates list without redundancy, however, there is always the concern of false positives, indeed, there was the need to establish a second filter. Once the redundant candidates are deleted, CLONA uses the second filter that eliminates false positives. This filter is applied to what we call to *partially* redundant entities. An entity is considered *partially* redundant if it belongs to two different mappings (*i.e.*, being given three ontological entities e_1, e_2 and e_3. If on the one hand, e_1 is aligned to e_2, and secondly, e_1 is aligned to e_3, this last alignment is qualified as doubtful. We note that CLONA generates $(1:1)$ alignments. To overcome this challenge, CLONA compares the topology of two suspicious entities (e_3 and its neighbor e_4) with respect to the redundant entity e_1 and retains the couple having the highest topological proximity. All candidates following the application of this filter is the subject of alignment file result.

Alignment Generation : The result of the alignment process provides a set of mappings, which are serialized in the RDF format.

1.2 Specific techniques used

CLONA has implemented a technique for determining alignment candidates across the power of Lucene search engine. In addition, during the translation phase, we have set up a local translator that is built during the alignment process. This treatment reduces the translation time cost and access to the external resource.

1.3 Link to the system and parameters file

CLONA is an open source ontology matching system and is available through this link (`http://www.mediafire.com/download/f6tacrt82sx316u/CLONA_OAEI_2015.zip`).

2 Results

Our system CLONA has been developed with a unique focus on multilingual ontologies the processing, through Multifarm test base. This dataset is composed of a subset of the Conference track, translated in nine different languages (*i.e.*, Chinese, Czech, Dutch, French, German, Portuguese, Russian, Spanish and Arabic).

3 General Comments

CLONA obtained an F-measure average of 43% and this, positions it in the second place among methods of the OAEI 2015 campaign. The translation treatment has been successful, especially with the technique of pivot language that reduces all ontological entities to one language, which is English. In addition, the enrichment with WordNet as an external resource, increased produced alignments accuracy. The evaluation was conducted according to two scenarios, as shown in Table 3. The first scenario is significantly better than the second, this is explained by the fact that ontologies share the same structure. Indeed, the structural similarity for ontological entities will be important. These values positioned CLONA in the second place compared to OAEI 2015 participant methods. It should be emphasized that in the case Same Ontologies, and over 45 treated language pairs, CLONA ranked first out of 15 couples. This performance is achieved thanks to the Recall values, which reflect the accuracy of the obtained alignments even in the cross-lingual context [3].

Table 1. F-measure average value for CLONA on Multifarm track for both test scénaios (Same Ontologies and Different Ontologies)

	Same Ontologies	Different Ontologies
	F-measure	**F-measure**
CLONA	**0.58**	**0.39**

[3] More details are available on this link : http://oaei.ontologymatching.org/2015/results/multifarm/index.html

4 Conclusions

CLONA participation in OAEI 2015 was encouraging, as it supplies good F-measure values in the two considered scenarios. Results reflects some strengths and some positive aspects.

References

1. Benjamins, V., Contreras, J., Corcho, O., Gómez-Pérez, A.: Six challenges for the semantic web. In: Special Interest Group on Semantic Web and Information Systems (SIGSEMIS Buelletin). (2004)
2. Euzenat, J., Shvaiko, P.: Ontology Matching (Second Edition). Springer-Verlag, Heidelberg (DE) (2013)
3. Kachroudi, M., Ben Moussa, E., Zghal, S., Ben Yahia, S.: Ldoa results for oaei 2011. In: Proceedings of the 6th International Workshop on Ontology Matching (OM-2011) Colocated with the 10th International Semantic Web Conference (ISWC-2011), Bonn, Germany (2011) 148–155
4. Zghal, S., Kachroudi, M., Ben Yahia, S., Mephu Nguifo, E.: OACAS: Ontologies alignment using composition and aggregation of similarities. In: Proceedings of the 1st International Conference on Knowledge Engineering and Ontology Development (KEOD 2009), Madeira, Portugal (2009) 233–238
5. Euzenat, J., Ferrara, A., Meilicke, C., Pane, J., Scharffe, F., Shvaiko, P., Stuckenschmidt, H., Sváb-Zamazal, O., Svátek, V., dos Santos, C.T.: Results of the ontology alignment evaluation initiative 2010. In: Proceedings of the 5th International Workshop on Ontology Matching (OM-2010), Shanghai, China, November 7, 2010. Volume 689 of CEUR-WS. (2010)
6. Euzenat, J., Ferrara, A., van Hage, W.R., Hollink, L., Meilicke, C., Nikolov, A., Ritze, D., Scharffe, F., Shvaiko, P., Stuckenschmidt, H., Sváb-Zamazal, O., dos Santos, C.T.: Results of the ontology alignment evaluation initiative 2011. In: Proceedings of the 6^{th} International Workshop on Ontology Matching (OM-2011), Bonn, Germany, October 24, 2011. Volume 814 of CEUR-WS. (2011)
7. Kachroudi, M., Ben Yahia, S., Zghal, S.: Damo - direct alignment for multilingual ontologies. In: Proceedings of the 3^{rd} International Conference on Knowledge Engineering and Ontology Development (KEOD), 26-29 October, Paris,France (2011) 110–117
8. Ngo, D., Bellahsene, Z.: Yam++ results for oaei 2012. In: Proceedings of the 9^{th} International Workshop on Ontology Matching (OM-2012) Colocated with the 11^{th} International Semantic Web Conference (ISWC-2012). Volume 946 of CEUR-WS., Boston, USA (2012) 226–233
9. Groß, A., Hartung, M., Kirsten, T., Rahm, E.: Gomma results for oaei 2012. In: Proceedings of the 9^{th} International Workshop on Ontology Matching (OM-2012) Colocated with the 11^{th} International Semantic Web Conference (ISWC-2012). Volume 946 of CEUR-WS., Boston, USA (2012) 133–140
10. Kachroudi, M., Zghal, S., , Ben Yahia, S.: When external linguistic resource supports cross-lingual ontology alignment. In: In Proceedings of the 5^{th} International Conference on Web and Information Technologies (ICWIT 2013), 9-12, May, Hammamet, Tunisia (2013) 327–336

CroMatcher - Results for OAEI 2015

Marko Gulić [a], Boris Vrdoljak [b], Marko Banek [b,c,1]

[a] Faculty of Maritime Studies, Rijeka, Croatia
marko.gulic@pfri.hr
[b] Faculty of Electrical Engineering and Computing, Zagreb, Croatia
boris.vrdoljak@fer.hr
[c] Ericsson Nikola Tesla d.d., Krapinska 45, HR-10000 Zagreb, Croatia

Abstract. CroMatcher is an ontology matching system based on parallel composition of basic ontology matchers. There are two fundamental parts of the system: first, automated weighted aggregation of correspondences produced by different basic matchers in the parallel composition; second, an iterative final alignment method. This is the second time CroMatcher has been involved in the OAEI campaign. Basic improvement with respect to the previous version has been implemented in order to speed up the system.

1 Presentation of the system

CroMatcher is an automatic ontology matching system for discovering correspondences between entities of two different ontologies. This is the second version of the system. The first version [1] was presented in the OAEI campaign held in 2013. In this second version, the system architecture remained unchanged but the system implementation was modified as well as the implementation of several basic matchers in order to speed up the system. Our goal was to prepare the system for the following test sets: Benchmark, Anatomy, Conference and Large Biomedical Ontologies. The system is fully prepared for the Benchmark, Anatomy, and Conference. It is partly prepared for the Large Biomedical Ontologies (only for the 10% fragments of ontologies). We are currently working to speed up our system even more and we expect to present it in the next OAEI campaign.

1.1 State, purpose, general statement

As stated before, the architecture of the new version of the system remained unchanged according to the first version [1] from 2013. To recapitulate, CroMatcher contains several terminological and structural matchers connected through sequential-parallel

[1] Presently at Ericsson Nikola Tesla, the research was done while working at the University of Zagreb

composition. First, the terminological basic matchers are executed. These matchers are connected through a parallel composition. After the execution of terminological matchers, the weighted aggregation is performed in order to determine the aggregated correspondence results of these matchers. These aggregated results are used in the execution of the structural matchers as initial values of entity correspondences. Structural matchers are also executed independently of each other in another parallel composition. Again, weighted aggregation is performed in order to determine the aggregated correspondence results of the structural matchers. Before the final alignment, the aggregated correspondence results of the terminological matchers and the aggregated correspondences' results of the structural matchers need to be aggregated using weighted aggregation. Eventually, the method of the final alignment is executed. This method iteratively takes the best correspondences between two entities into the final alignment.

1.2 Specific techniques used

In this section, only the modified components will be described in detail. The rest of the main components are described in the first version of the system [1]. We modified some terminological and structural matchers in order to speed up the matching process. These matchers are modified for the test sets Anatomy and Large Biomedical Ontologies because the ontologies in these test sets contain a large number of entities. Our matcher first counts the number of entities. If the ontologies contain more than 1000 entities than the modified versions of some matchers are activated instead of the original versions of matchers. Furthermore, we modified one terminological basic matcher in order to read entity information from components *oboInOwl#hasRelatedSynonym* and *oboInOwl#hasDefinition*. These components are implemented within ontologies of the Anatomy test set and contain considerable information about entities. The modified basic matchers are the following:

1. **Terminological matchers:**

- Matcher that compares ID and annotation text of two entities (classes or properties) with the n-gram matcher [2] is extended in a way that also compares the text obtained from components *oboInOw#hasRelatedSynonym* and *oboInOwl#hasDefinition*. As stated before, these components are implemented within ontologies in the Anatomy test set. Our system first checks whether these components are implemented. If these components are not implemented within compared ontologies, the matcher compares only the ID and annotations like before.
- Matcher that compares textual profiles of two entities with TF/IDF [3] and cosine similarity [4] is modified for the ontologies that contain more than 1000 entities in order to speed up the matching process. A textual profile is a large text that describes an entity (text obtained from annotations of compared entity and its all sub entities) therefore the matching was very slow because the TF/IDF method need to load the text of all entities before starting comparing two entities. When a target ontology contains more than 1000 entities, a modified implemented matcher is activated. This matcher compares textual profiles of two entities with the string metric described in [5]. This metric calculates similarity based on

adjacent character pairs that are contained in both strings. This string metric is much faster than the TF/IDF method but the matching results are a bit worse than the results obtained with TF/IDF method. It is acceptable because the system performs the matching process faster enough to match ontologies with many entities.

- Matcher that compares individuals of two entities by applying TF/IDF and cosine similarity is modified for the ontologies that contain more than 1000 entities. If the ontology contain more than 1000 entities, a modified implemented matcher with string metric described in [5] is activated like in the previous basic matcher.
- Matcher that compares extra individuals of two entities with TF/IDF and cosine similarity is modified like two previous matchers in order to speed up the matching process.

2. **Structural matchers:**

- All structural matchers described in the first version of our system [1] are executed iteratively. In order to speed up the matching process, we also made modification when comparing ontologies that contain more than 1000 entities. All structural matchers are executed just once (instead of being executed iteratively many times) when comparing the ontologies with more than 1000 entities. This speeds up the matching process but decreases the quality of matching process when comparing large ontologies. In the next version of the system, our major concern will be to solve the problem of slow iterative execution of structural matchers.

2 Results

In this section, the evaluation results of CroMatcher matching system executed on the SEALS platform are presented.

2.1 Benchmark

In OAEI 2015, Benchmark includes two test sets: Biblio and Energy. In Table 1 the results obtained by running the CroMatcher ontology system can be seen.

Table 1. CroMatcher results for Benchmark test set

Test set	Recall	Precision	F-Measure	Time (s)
Energy	0.21	0.96	0.67	-
Biblio	0.82	0.94	0.88	485

The result for Biblio test set is equal to the result obtained at the OAEI 2013 campaign because the actual system is equal to the previous version of our system when the system matches ontologies that have less than 1000 entities. The execution time for Biblio test set was reduced by 50%, which is the result of the optimization of the program code. Our system achieves the best result in this test set together with the Lily system (F-measure 0.88). The Energy test set is new Benchmark test set. Our system achieves the third best result for this test set. Given the overall results of these two test

sets, our system achieves the best result for the Benchmark test set. Most of the ontologies in Benchmark test set are implemented without entity annotations (label and comment) therefore it can be concluded that our system uses well the information from other ontology components in order to find alignment between two ontologies.

2.2 Anatomy

In OAEI 2015, the Anatomy test set consist of two large ontologies (mouse.owl and human.owl) that have to be matched. In Table 2 the results obtained by running the CroMatcher ontology system can be seen.

Table 2. CroMatcher results for Anatomy test set

Test set	Recall	Precision	F-Measure	Time (s)
Anatomy	0.814	0.914	0.861	569

Our system achieves the sixth best result for this test set. The result of our system (F-measure 0.861) is very close to the results of the better systems in this test set except the result of the system AML which is the only system with F-measure greater than 0.9 (0.944). The result for Anatomy test set is a bit lower than we expected. It is lower because the system activates modified basic matchers for the ontologies with more than 1000 entities and these matchers (especially non-iterative structure matchers) are not as good as the original basic matchers but they speed up the system very much. In OAEI 2013, our system did not finish to match ontologies in the Anatomy test set even after 5 hours which was the time limit for the OAEI 2013 campaign. Therefore, a little bit lower result is, in our opinion excusable in exchange for the speed of execution. However, a remaining challenge for future work is to speed up the execution of the iterative structural matcher in order to improve the matching results for Anatomy test set. Also, we have to improve the usage of the information obtained by components *oboInOwl#hasRelatedSynonym* and *oboInOwl#hasDefinition* which are not the standard component of the OWL ontology but are the standard implemented components in mouse.owl and human.owl ontologies.

2.3. Conference

In OAEI 2015, Conference test set consist of 16 small ontologies that have to be matched to each other. In Table 3 the results obtained by running the CroMatcher ontology system can be seen.

Table 3. CroMatcher results for Conference test set

Test set	Recall	Precision	F-Measure	Time (s)
Conference	0.50	0.59	0.54	183

The result for Conference test set classifies our system among the worst ontology systems for this test set. These ontologies mutually have approximate about ten exact correspondences therefore the best matching systems found about two correspondences more than our system which is not the big difference but considering the results of the Benchmark test set, we expected to have better result. Considering the implementation

of these ontologies, it can be seen that all entities have the meaningful ID or label which is not the case for Benchmark test set. Therefore, in the Benchmark test set the threshold of the final alignment has low value but in Conference test set where all entities have meaningful names, we believe that the threshold needs to be higher. This is obviously one more challenge for the next version of our system.

2.4. Large Biomedical Ontologies, Multifarm, Interactive, Ontology Alignment for Query Answering and Instance matching

The system had problems with Large Biomedical Ontologies therefore we have to speed it up more before the next evaluation. For other test sets (Multifarm, Interactive, Ontology Alignment for Query Answering and Instance matching) the matching process itself needs to be modified and we did not prepare the system for these test sets.

3 General comments

We are very pleased for the opportunity to evaluate our ontology matching system on the SEALS platform and thus compare our system with other existing systems. There are many different test cases and we think that these test cases will help us make additional improvements of our system in the future.

3.1 Comments on the results

Our system shows great results in Benchmark test set again. We can be satisfied with the result of Anatomy test set but we will try to improve the system for these test sets. Moreover we will make our system capable of processing the sets for which we did not prepared it in this campaign.

3.2 Discussions on the way to improve the proposed system

We applied faster measure than TF/IDF to compare different documents of entities. We will try to solve the problem with the slow iterative structural matcher. Also, we will have to store the data about the entities in a separate file instead of java objects in order to reduce the usage of memory in the system.

4 Conclusion

The second version of the CroMatcher ontology matching system and its results were presented in this paper. The evaluation results show that CroMatcher achieved considerable results for Benchmark and Anatomy test sets. The matching process is executing much faster than the matching process in the first version of the system but there is still room for improvement considering speed of the process. Also, the system

needs to be modified for the special test sets in the OAEI campaign like Instance matching or Multifarm. We will try to solve these problems and prepare the system to be competitive in all OAEI test sets next year.

References

1. Gulić, M., Vrdoljak, B.: CroMatcher - results for OAEI 2013, Proceedings of the 8th International Workshop on Ontology Matching, pp. 117–122, Sydney, Australia, 2013.
2. Euzenat, J., Shvaiko, P.: Ontology matching. Springer, 2007.
3. Salton, G., McGill, M.H.: Introduction to Modern Information Retrieval. McGraw-Hill, New York (1983)
4. Baeza-Yates, R., Ribeiro-Neto B.: Modern Information Retrieval. Addison-Wesley, Boston (1999)
5. Strike a match, http://www.catalysoft.com/articles/strikeamatch.html, accessed: 06.10.2015.

DKP-AOM: results for OAEI 2015

Muhammad Fahad
DISP Lab (http://www.disp-lab.fr), Université Lumière Lyon2
160 Boulevard de l'Université, Bron, FRANCE
firstname.lastname@univ-lyon2.fr

Abstract
In this paper, we present the results obtained by our DKP-AOM system within the OAEI 2015 campaign. DKP-AOM is an ontology merging tool designed to merge heterogeneous ontologies. In OAEI, we have participated with its ontology mapping component which serves as a basic module capable of matching large scale ontologies before their merging. This is our first successful participation in the Conference, OA4QA and Anatomy track of OAEI. DKP-AOM is participating with two versions (DKP-AOM and DKP-AOM_lite), DKP-AOM performs coherence analysis and has no consistency principle violation. In OA4QA track, DKPAOM out-performed in the evaluation and generated accurate alignments allowed to *answer all the queries* of the evaluation. Also, we can see its competitive results for the conference track in the evaluation initiative among other reputed systems. In the anatomy track, it has produced alignments within an allocated time and appeared in the list of systems which produce coherent results. Finally, we discuss some future work towards the development of DKP-AOM.

Keywords: Ontology matching, Ontology merging, disjoint knowledge, inconsistency, incompleteness, inconciseness, validation of mappings, verification of merged ontology

1 Presentation of the System

Ontology merging is a process of building a new ontology from two or more existing ontologies with overlapping parts. The merged ontology can be either virtual or physical, but must be consistent, coherent and include all the information from the source ontologies [1]. Ontology merging is based on two primary steps. Firstly, the source ontologies are looked-up for correspondences between them. Secondly, duplicate-free and conflict-free union of source ontologies is achieved based on the established correspondences [2]. The first part mainly comes under the ontology matching, whereas the second part targets to achieve the merged ontology based on the results of the first part, i.e., mappings between source ontologies. To produce accurate merged ontology, there should be some mechanism to avoid erroneous intermediate mappings and also to merge them in such a way that produces consistent, complete and coherent merged ontology. There are many hurdles that come across in the generation of desired merged output. Firstly, ontological errors and design anomalies that can occur in the source ontologies detract from reasoning and inference mechanisms, and create

bottleneck in their integration tasks [3]. In addition, conceptualization of domain, explication and modeling of knowledge over ontologies and semantic heterogeneities make their integration more difficult [4]. Secondly, even if the individual ontologies are free from errors, some of the identified mappings lead towards the erroneous situations producing several types of errors in the merged ontology [5]. For building an effective ontology merging algorithm, it is essential to incorporate ontological error checking during the validation of ontology mapping process and the verification of merged ontology to attain the accuracy of resultant output.

In order to meet the above mentioned challenges for the ontology merging research, we proposed semi-automatic DKP-OM system implemented in Jena framework for the merging of heterogeneous ontologies with the human user expert [6]. Later, we released a fully Automatic Ontology Merging (AOM) system named DKP-AOM implemented in OWLAPI 3 [7]. The name DKP comes from the concept of performing *Disjoint Knowledge Analysis (DKA)* and *Disjoint Knowledge Preservation (DKP)* during the merging process. Disjoint Knowledge Analysis plays a vital role in controlling the search space for finding similarities between source ontologies. Look-up within disjoint partitions of source ontologies significantly reduces the time complexity of the mapping phase. Disjoint Knowledge Preservation in the merged ontology helps to preserve disjoint axioms in the sub-hierarchies of merged ontology to avoid incompleteness in the resultant merged ontology. In this way, it also pin-points different conflicts between source ontologies based on disjoint axioms in the source ontologies and detects inconsistent mappings. Computed mappings that lead in many cases to a large number of unsatisfiable classes are eliminated so the resultant merged ontology should not suffer from inconsistencies. The next sub-sections provide more details about DKP-AOM and then discuss our results of OAEI participation.

1.1 State, purpose, general statement

Our system DKP-AOM follows a five step methodology as illustrated in the Figure 1. First, it generates the intermediate models (OWL-DL Graphs) of source ontologies and does preprocessing on the concept URIs and labels. Second using these graphs, *MatchManager* component performs the first level task of finding the initial linguistic, synonym and axiomatic based mappings between concepts. For this, it first builds the search space based on disjoint axioms inside the source ontologies for finding the correspondences between the ontologies.

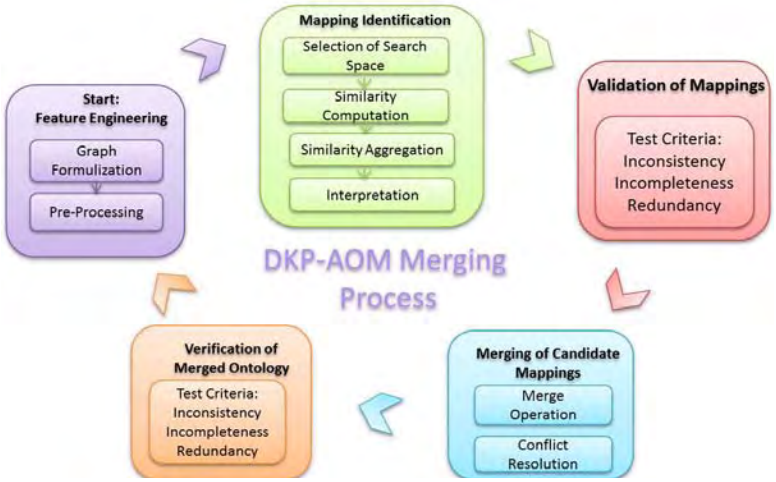


Figure 1. DKP-AOM merging process

MatchManager performs the following similarity measures to detect mappings:

Concept Similarity. It employs several basic matchers to find mappings between concepts. Concept URI similarity (Sim_{uri}) and Label similarity (Sim_{lab}) computes lexical and synonym based correspondences at the element level between source ontologies. Lexical similarity finds the string-based correspondences based on *SimMetric* [28]. Synonym similarity is computed based on the lexical database *Wordnet* [29] that helps to detect the concepts which have the same meanings but are lexically different.

Datatype and Object property Similarity. Correspondences between Datatype and Object properties are identified on the basis of their URI and Label similarities. It also considers domain and range assosicated with them to detect a perfect match.

Inheritance based Similarity. It also considers that an inheritance is a vital factor to detect the mapping candidates between source ontologies. It increases a level of confidence that the detected mappings have not just a lexical similarity, but a real mapping having parent-child relation as well. *Inheritance* matching is done after the *Concept label* similarity and *synonym* similarity. Consider a scenario, where an ontology O1 contains Person and PhD concepts. Person is defined as: *{Person subclassof hasName some String},* and concept PhD is defined as a subconcept of Person concept as: *{PhD subClass Person}*. In ontology O2, there is a PhD candidate that is defined as: *{PhD subclass of hasName some String}*. In such a case we get the basic mappings between O1:PhD, and O2:PhD. Then the inheritance matcher plays an important role by matching the inheritance of the restriction from Person to PhD from O1 with the restriction in O2 and adds the confidence level of their similarity. In this way, an inheritance matching has a potential impact in the proposed solution. This similarity will help for the detection of axiomatic similarities between concepts.

Concept DL Axiomatic Similarity (Sim_{axm}). OWL classes are described through the class descriptions/expressions that enrich the background information of the concepts and represent the constraints of real world situations. For finding the accurate semantic similarity between the concepts of ontologies, DL axioms can help significantly as they define the context of the concepts. They link the concept by different means that depict the concept's real semantics. Therefore, it gets axiomatic definitions, which can be formed from the union, complement, intersection and restriction operators applied on the primitive concept or the anonymous concept and/or by their boolean combinations, and performs matching to detect such DL axiom similarities. This is the most difficult part of matching, although most significant as well. Most ontologies in *Conference track* in OAEI were equipped with OWL DL axioms of various kinds, therefore it opens a way to use semantic matching.

MatchManager aggregates the individual similarities between ontologies and propagates the results to *ConsistencyChecker* for the validation of mappings. Third, *ConsistencyChecker* has many detectors that make the validation of each mapping found in the initial stage so that the merged ontology stays consistent with reference to the source ontologies. When the initial mappings pass the consistency test, *ConsistencyChecker* passes the mappings to the *Reasoner*.

Fourth, *Reasoner* aggregates the output of different similarity measures, resolves conflicts and merges mappings to generate a global merged ontology. This *Reasoner* is a component of DKP-AOM system and not an open source DL Reasoner engine. It implements various patterns (see detail in [7]) for the automatic merging of source ontologies in case of different types of conflicts and structural differences.

Finally, it compiles the output as a merged global ontology automatically or a final list of consistent mappings as required by the end user. Our merging algorithm imports the first ontology as the merged ontology and then performs several operations to build the combined definitions for each of the concepts from the source ontologies. Each of the axiomatic definitions from the source ontologies are matched together, merging is performed on them, and the *combined rich axioms* are added in the merged ontology. Our merging algorithm performs deletion of axioms or the rewriting of some of

them in order to preserve desired consequences while removing the undesired ones. Merging of axiomatic definitions really achieves a richer merged ontology which captures sufficient definitions from the source ontologies. Finally, it applies the quality criteria and ensures the ultimate goal of achieving the satisfiability of merged ontology by checking the correctness and consistency of concepts, properties, and axioms of the generated ontology.

1.2 Specific Techniques Used

Data Preprocessing: Linguistic analysis of concept labels and properties is done with the help of MorphAdorner* (version 1.0). MorphAdorner is helpful in various cases especially the lemmatization process is worth useful for detecting the base words of terms and irregular verbs used in source ontologies. For example, concept "students" to lemma "student" and properties ("Accepted", "Accepting", "Accept", and "Accepts") to their base "accept". MorphAdorner is really helpful for the detection of similarities between properties which are usually not in the base form.

Search Space Analysis based on Disjoint Partitions: It is very important to build the search space for the lookup of mappings between ontologies. In general, it requires exhaustive analysis (or complete comparison) for the similarity computation between the concepts of ontologies, where each concept *c* of the ontology O_a is matched with each concept *c'* of the ontology O_b. The restriction of look-up with in disjoint partitions minimizes the search space for the mapping computation [8]. For an example, consider conference ontologies illustrated in Figure 2. There are 14 concepts in *O1:CRS_DR* ontology and 36 concepts in *O2:CMT ontology*. In *CRS_DR* ontology, three disjoint axioms between level-1 concepts, partition the domain concepts in four non-overlapping domains, i.e., *Program, Person, Document and Event*. *CMT* ontology partitions the concept into six disjoint categories, i.e., *Person, Decision, Document, Conference, Preference*, etc. These ontologies allow the concept mapping search space look-up within disjoint partitions. For example, search spaces look-up within (*O1:Person, O2:Person*), (*O1:Document, O2:Document*), (*O1:Event, O2:Conference*). For concept matching, it needs 14*36 = 504 comparisons. But, lookup in disjoint partitions makes the search space much smaller, and requires maximum 155 comparisons (by manual calculation) for mapping the concepts of these ontologies.

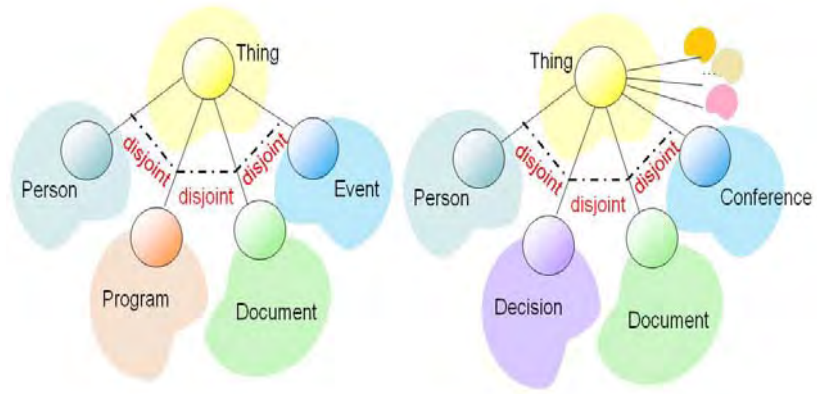


Figure 2. Top level Disjoint Partitions in CRS_DR and CMT conference ontologies

We believe this idea of *Divide and Conquer* is helpful for the large ontologies. The more disjoint axioms are modeled in the input ontologies, the more search space is restricted, which impacts in

* http://morphadorner.northwestern.edu

lower number of comparisons for the identification of mappings. We call this disjoint based partitioning strategy as *Divide and Conquer* approach. In fact, it resembles, but not achieves a full or successful conquer in all scenarios of partitioning, as divide and conquer strategies found in an algorithm domain where the divided partition is at last conquered with success. For example, in case of failure, when a concept does not found its mapping concept in the divided partition, it is matched with the other level of concepts to find its mapping (just like exhaustive search in the entire space but step-by-step).

Validation of Mappings: For the evaluation of ontologies, Gomez-Perez constructed an error taxonomy as a guideline for ontology engineers to help building well-formed and well-structured classification of concepts in the ontologies. She defined three classes of ontological errors that might occur when modeling the conceptualization into taxonomies, i.e., *Inconsistency, Incompleteness and Redundancy* [9] as illustrated in Figure 3. Inconsistency in ontology means that there is some *sort of contradictory knowledge* inferred from the concepts, definitions and instances within the ontology. It creates ambiguity, contradictions in interpretations and compromises precision of results. Incompleteness occurs when ontologists model the domain knowledge in the form of concepts, properties and definitions, but *overlooked some of the important information* about the domain. The incompleteness of domain knowledge lacks reasoning and prevents inference mechanisms. The other important task is to make ontologies concise *without repeating and replicating same information* so that they store only necessary and sufficient knowledge about the concepts, axioms and properties. Redundancy errors not only compromise conciseness and usability, but also create problems for the maintenance and manageability of ontologies. We used this framework as a test criteria for the validation of mapping and verification of merged ontology [10]. Computed initial mappings that lead to unsatisfiable classes in merged ontology are eliminated so the resultant merged ontology should not suffer from inconsistencies. This test criteria serves best for the detection of inconsistent mappings and also for ensuring the satisfiability of a merged ontology.

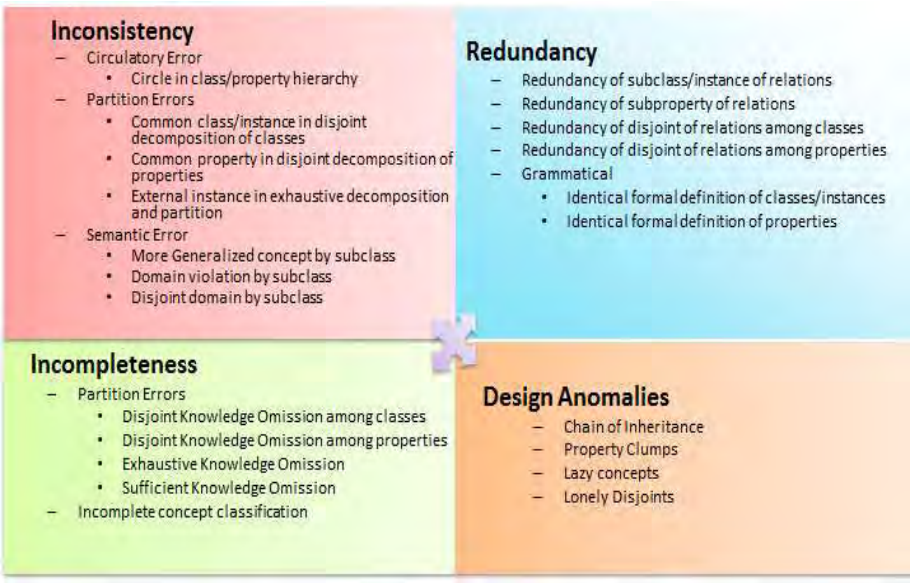


Figure 3. Test criteria for the validation of mappings and verification of merged ontologies

1.3 Adaptations made for the evaluation

As you read above, DKP is an automatically merging system. Therefore it was developed based on user GUIs such as source ontology trees for display, visual alignments between ontologies, merged ontology tree, etc. The original version of DKP has changed and these visual components are removed so that it can participate under the seals platform. However, still it needs proper clean-up to improve its runtime for the future OAEI participations.

1.4 Link to the system

Various versions of my system can be found at my personal site: http://sites.google.com/site/mhdfahad under *plugins* tab. The mapping system is separated from the merging system, and can be downloaded according to needs. For the merging of ontologies, use the same command of seals platform with –o following three paths, two for source ontologies and one for the output merged ontology. As a result of this command, a list of ontology mappings and a resultant merged ontology are produced.

2 Results

In order to show the efficiency and effectiveness of our system, this year we participated in Conference and Anatomy tracks. The results are very encouraging provided by the OAEI 2015 campaign as our system is acceptable and comparable with other participants, and are discussed in the following subsections.

2.1 Conference

The goal of conference track is to find alignments among 16 ontologies relatively smaller in size (between 14 and 140 entities) but rich in semantic heterogeneities about the conference organization domain. As a result, Alignments are evaluated automatically against reference alignments. Therefore, it is very interesting to measure the Precision, Recall and F-measure of our system on ontologies rich in OWL DL axioms of various kinds, and also does a comparison between existing systems to see their performance on real world datasets. The resultant match quality was evaluated against the original (ra1) as well as entailed reference alignment (ra2) and violation free version of reference alignment (rar2). We achieved F-Measure values better than the two Baselines results (edna, StringEquiv). Table 1 presents the results obtained by running DKP-AOM on the Conference track of OAEI campaign 2015. Our system DKP-AOM has produced very competitive results among top ranked systems. Our precision measure is significantly high, recall is good, giving comparable F-measure value to depict a real effort towards detecting heterogeneities for the goal of ontology matching.

alignments	Precision	F.5-measure	F1-measure	F2-measure	Recall
ra1-M1	0.84	0.77	0.69	0.63	0.59
ra1-M3	0.84	0.74	0.63	0.54	0.5
ra2-M1	0.79	0.72	0.63	0.57	0.53
ra2-M3	0.79	0.69	0.57	0.49	0.45
rar2-M1	0.78	0.72	0.65	0.58	0.55
rar2-M3	0.78	0.69	0.59	0.51	0.47

Table 1. DKP-AOM results on conference track ontologies

DKP-AOM has given excellent performance for the evaluation based on the logical reasoning where oaei competition applied detection of conservativity and consistency principles violation. Our DKP-AOM is among five best tools which have no consistency principle violation (see Table 2), as we have employed various algorithms for the validation of initial mappings.

The lowest number of conservativity principle violations has LogMap-C which has a repair technique for them. DKP-AOM has produce second-lowest number of conservativity principle violations, and employed algorithms to maintain conciseness and avoid redundancies in the resultant ontology. Conservativity principle violations can be favored by redundancies, but those are not the only source of violations, due to possible complex interactions with other axioms in both ontologies. Further four tools have average of conservativity principle around 1.

Matcher	#unsat. onto	#align	#incoh. align	#totalConser. Viol.	#avgConser. Viol.	#totConsist Viol.	#avgConsist. Viol.
LogMap-C	0	21	0	5	0.24	0	0
DKP-AOM	**0**	**21**	**0**	**16**	**0.76**	**0**	**0**
XMAP	0	21	0	19	0.9	0	0
JarvisOM	0	21	2	27	1.29	7	0.33
LogMap	0	21	0	29	1.38	0	0
AML	0	21	0	39	1.86	0	0

Table 2. Statistics of consistency and conservativity principle violations

2.2 Ontology Alignment for Query Answering (OA4QA)

In this track, our system has out-performed and generated excellent results (see Table 3). Precision and recall has calculated with respect to the ability of the generated alignments to answer a set of queries in an ontology-based data access scenario where several ontologies exist. AML, DKPAOM, LogMap, LogMapC and XMap were the only matchers whose alignments allowed to answer **all the queries** of the evaluation. The best global results have been achieved for violations queries, that has been correctly covered by AML, DKP-AOM, LogMap, COMMAND, LogMapC and XMAP, in decreasing order of f-measure w.r.t. RA1. Notably, DKP-AOM achieved an impressive *f-measure* of 0.999 w.r.t. RAR1, showing an effective handling of logical violations.

Matcher	Successful queries	Precision (RA1)	Recall (RA1)	F-Measure (RA1)	Precision (RAR1)	Recall (RAR1)
Global Evaluation Results						
DKPAOM	18/18	0.667	0.618	0.635	0.666	0.639
Advanced Queries Results						
DKPAOM	5/5	0.200	0.100	0.133	0.200	0.100
Basic Queries Results						
DKPAOM	6/6	0.667	0.667	0.667	0.667	0.667
Violations Queries Results						
DKPAOM	7/7	1.000	0.947	0.967	0.999	1.000

Table 3. DKP-AOM results on Ontology Alignment for Query Answering (OA4QA) track

2.3 Anatomy

The anatomy real world case is about matching two very large biomedical ontologies, i.e., Adult Mouse Anatomy (2744 classes) and the NCI Thesaurus (3304 classes) describing the human anatomy.

We participated with two versions DKP-AOM and DKP-AOM_lite, DKP-AOM performs coherence analysis. The evaluation was run on a server with 3.46 GHz (6 cores) and 8GB RAM, with allocated time less than an hour. This year 2015, there are 11 different systems (not counting different versions) which generated an alignment, out of them only four systems participated in the anatomy track for the first time. These are COMMAND, GMap, JarvisOM and DKP-AOM (with two version). Two of them COMMAND and GMap run out of memory and could not finish execution with the allocated amount of memory, therefore, their execution times are not fully comparable to the other systems. Our systems have produced results within an allocated time, illustrated in Table 4 with other systems.

Importantly, our DKP-AOM achieves coherency and became in the list of 7 systems which produced only coherent mappings. It has also generated only trivial correspondences.

F-Measure of DKP-AOM_lite (0.763) is very near to the baseline which is based on (normalized) string equivalence (StringEquiv, 0.766), with difference of only .003.

Matcher	Runtime	Size	Precision	F-Measure	Recall	Recall+	Coherent
COMMAND	63127*	150	0.293	0.053	0.029	0.042	x
GMap	2362**	1344	0.916	0.861	0.812	0.534	--
JarvisOM	217	458	0.365	0.169	0.11	0.01	-
DKP-AOM	370	201	0.995	0.233	0.132	0.0	x
DKPAOM-lite	476	949	0.991	0.763	0.62	0.042	-
StringEquiv	-	946	0.997	0.766	0.622	0.000	-

Table 4. Results of first time participating systems on Anatomy track

3 Conclusion and Future Directions

The participation of DKP-AOM in OAEI 2015 is a success in the conference and OA4QA track. In OA4QA track, DKPAOM out-performed in the evaluation and generated accurate alignments which allowed to answer all the queries of the evaluation. For the Conference track, DKP-AOM has given excellent performance for the evaluation based on the logical reasoning where oaei competition applied detection of conservativity and consistency principles violation. It showed a real effort as it has *no consistency principle violation*, and also has produce *second-lowest* number of conservativity principle violations. It also presented intermediate results in the Anatomy track and comes in the list of 7 matching system which produce coherent results. However, the whole framework of DKP-AOM is very huge and the participated version needs more effort of development to achieve more success in the upcoming OAEI. We plan to integrate synonym based mappings in the participated version. In addition, we plan to implement all the test criteria inside the DKP-AOM and present it as a complete system that achieves consistency, completeness and coherency.

Our technique of building search space is based on the disjoint partitions available in the source ontologies (that are very rarely present in the dataset ontologies). One of our future directions is to devise an disjoint learning algorithm to identify and make disjoint partitions automatically in the source ontologies, even if disjoint partitions were not present in the source ontologies before their merging.

References

1. Bruijn, J.d., Ehrig, M., Feier, C., Martín-Recuerda, F., Scharffe, F., and Weiten., M., Ontology mediation, merging and

aligning. In Semantic Web Technologies. Wiley 2006

2. Euzenat, J., and Shvaiko, P., Ontology Matching. Springer, 2007, ISBN 978-3-540-49611-3.
3. Fahad, M., Qadir, M.A., Noshairwan, M.W., Ontological Errors - Inconsistency, Incompleteness and Redundancy. In Proceedings of 10th Intl Conference on Enterprise Information Systems, pp. 253-285, 2008, Spain, Springer,
4. Klein, M., (2001): Combining and relating ontologies: an analysis of problems and solution. In Proc. of Workshop on Ontologies and Information Sharing (IJCAI), pp. 53-62. Seattle, USA (2001)
5. Fahad, M., and Qadir, M.A., A Framework for Ontology Evaluation, 16th ICCS Supplement Proceeding, vol. 354, 2008, France, pp.149-158.
6. Fahad, M., Qadir, M.A., Noshairwan, W., Iftakhir, N., DKP-OM: A Semantic based Ontology Merger, Proceedings of 3rd International Conference on Semantic Technologies (I-Semantics 07) Graz, Austria, 2007, Pages 313-322
7. Fahad, M., Moalla, N., Bouras, A., Detection and Resolution of Semantic Inconsistency and Redundancy in an Automatic Ontology Merging System, Journal of Intelligent Information System (JIIS), Vol. 39(2) pp. 535-557, 29/4/2012, DOI 10.1007/s10844-012-0202-y
8. Fahad, M., Moalla, N., Bouras, A., Qadir, M.A., Farukh, M., Disjoint Knowledge Analysis and Preservation in Ontology Merging Process, proceedings of 5th International Conference on Software Engineering Advances (ICSEA'10), IEEE CS, August 22-27, 2010 - Nice, France.
9. Gómez-Pérez, A., (2001): Evaluating ontologies: Cases of Study. IEEE Intelligent Systems and their Applications, vol. 16(3): 391–409, (2001)
10. Fahad, M., Moalla, N., Bouras, A., Towards ensuring Satisfiability of Merged Ontology, International conference on computational science, ICCS 2011, Procedia Computer Science 4 (2011), pp. 2216–222, 1-3 june, 2011

EXONA Results for OAEI 2015

Syrine Damak, Hazem Souid, Marouen Kachroudi
and Sami Zghal

Université de Tunis El Manar, Faculté des Sciences de Tunis, LIPAH Programmation Algorithmique et Heuristique, 2092, Tunis, Tunisie;
damaksyrine@gmail.com
swdhazem@gmail.com
marouen.kachroudi@fst.rnu.tn
sami_zghal@planet.tn

Abstract. This paper presents the results of EXONA in the Ontology Alignment Evaluation Initiative (OAEI) 2015. EXONA is an automatic instance-based ontology alignment systems in which we parse ontology as first step. In the second step, we index instances of the first ontology. These indexed instances will be applied for the querying phase. In the last step, our system aligns instances based by aggregating score of different terminological matchers. We first describe the overall framework of our matching System (EXONA) then we detail the techniques used in the framework for instance matching. Last, we give a thorough analysis on our results and discuss some future work on our system. It's our first participation in the OAEI instance matching, the results are good in terms of recall, precision and F-measure.

1 Presentation of the system

Ontology matching is a key interoperability enabler for the semantic web, as well as a useful tactic in data integration tasks. Knowledge about one object may be contained in multiple and different knowledge bases. Therefore, a lot of work has already been built to obtain more complete knowledge about things existing in different domains. This is in order to exceed the area of divergence obstacle, by creating cross-domain knowledge.

Accordingly, it's strongly recommended to focus on the more active element of ontology which it called instance. Many instances matching approaches have been proposed, and among which is ours. In fact, our system is proposed for large scale instance matching. It operates on three successive modules, namely : transformation, indexation and correspondence. Transformation consists in transforming separately both of knowledge bases on an exploitable form and then creating our own instance object as a profession object. Indexation is the process of indexing instances of knowledge base; only instances of the source base knowledge

have to be indexed. Correspondence consists in querying the index already built. This request contains instances of the target knowledge base non indexed.

In order to solve the problem of large knowledge bases, we propose an index by concept to minimise the area and the time expended on searching instances behind the request technique.

1.1 State, purpose, general statement

This section describes the overall framework of EXONA . Our system includes three modules, i.e., *transformation, indexation and correspondence*.

The system proposed operates in three successive modules, each of those is branched into two phases. The system begins with transforming the knowledge bases into two independent graphs, those graphs formed by OWL nodes. After having constructed these graphs, it's time instance objects be created. The construction of instance object appealed the neighbourhood technical in which neighbourhood spread by similarity is done. This technique aims to enrich instance object by neighbouring instances with which a high similarity exists. We proceed after that by a terminological normalization of instances. This normalisation is compulsory for the indexation of instances as well as for the similarity calculation. We index after that instances of the source data knowledge. It 's not a blind indexation. In fact it is done by concept. This is in order to provide optimal search fields oriented concept. After index creation, it's time to query this index. The request emitted contains instances from the target knowledge base. Those instances have to be normalized before being passed through the request. After querying the index, a candidate set is returned. Each candidate pair is accompanied by Ã score indicating its rank behind the rest of pair. To identify the pair of instance that have to be aligned, we filter this set of candidate set by saving only the two best pairs. As a verification process, we calculate terminological similarity. This latter want to be combined by the score given on requesting phase. Those similarities are aggregated then to identify the pair of instances to be aligned having the higher similarity score.

1.2 Specific techniques used

The process of EXONA system consists in the following three successive modules, namely : transformation, indexation and correspondence.

1. **Transformation module**
   This module is branched into two phases, namely : graph construction and instance creation.
   - *Graph construction* : As input, our system receive two OWL files. Those files are transformed into two independent OWL graphs. Graphs are more adequate representation ensuring highlighting of information.
   - *Instance creation* : Instance object is an object formed by an identifier and a content. It is identify by an URI; its content is formed by a set of information which makes it an autonomous entity. It contains the list

of neighbouring instances spread by similarity. Moreover, it contains the RDF triplet of this one. To calculate similarity, we have used $Edit - distance$.

2. **Indexation module**

   This module is formed by two phases, namely : pretreatment and instance creation.
   - *pretreatment* : In this phase we remove special symbols like "Â£.*-", etc. and stop words like "the", etc. standardization of case, etc. This pretreatment serves then in the requesting phase.
   - *Indexation* : indexation aims to index instances of the source data knowledge. Each document is identfied by the URI of the instance and contains the content of instance formed only by its data property and the data property of the set of neighbouring instances spread by similarity.

3. **Correspondence module**

   This module is divided into two phases, namely : querying and Filtering and matches identification.
   - *Querying* : querying phase has as input instances of the target knowledge base. Those instances have to be pretreated before the process of research on the index. This phase generates as an output a list of candidates accompanied by their score.
   - *Filtering and matches identification* : During this phase, the system takes the two candidates having the highest score. Then, it calculates terminological similarity of this pair of candidate. EXONA system identifies the pair of instances to be aligned by aggregating similarities with the one given by the search process. The pair of instance to be aligned is the one having the best score.

1.3 Adaptations made for the evaluation

We have changed the version of Lucene from 4.10.2 to 3.6. In fact, our first one needs specific adaptations to be accepted by 2015 evaluation campaign.

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

http://www.mediafire.com/download/b3vx3zio02br45y/$EXONA_OAEI_2015$.zip

2 Results

The instance matching 2015 track contains two subtasks. Each task is articulated in two tests with different scales (i.e., number of instances to match): i) Sandbox (small scale). It contains two datasets called source and target as well as the set of expected mappings (i.e., reference alignment). ii) Mainbox (medium scale).

2.1 Author Disambiguation Task

The goal of the Author Disambiguation Task is to link OWL instances referring to the same person (i.e., author) based on their publications. This task is done with the two datasets previously invoked.

Sandbox task The Sandbox test aims to evaluate behaviour of our system with Ã small scales. Table 1 below presents the results obtained by running EXONA on the instance matching track of OAEI campaign 2015 done with the Sandbox task.

	Precision	**Recall**	**F-measure**
EXONA	0.941	0.941	0.941
InsMT+	0.834	0.705	0.764
Lily	0.981	0.981	0.981
LogMap	0.994	0.906	0.948
RiMOM	0.929	0.929	0.929

Table 1. Results of Author Disambiguation Task for Sandbox task

Mainbox task This task is also done with Mainbox task. The gaol of this test is to evaluate the behaviour of our system in large scale. Table 2 below presents the results obtained by running EXONA on the instance matching track of OAEI campaign 2015 done with the Mainbox task.

	Precision	**Recall**	**F-measure**
EXONA	0.0	0.0	NaN
InsMT+	0.76	0.665	0.709
Lily	0.964	0.964	0.964
LogMap	0.996	0.831	0.906
RiMOM	0.911	0.911	0.911

Table 2. Results of Author Disambiguation Task for Mainbox task

2.2 Author Recognition Task

The goal of Author Recognition 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. This task is done also with the two datasets previously invoked.

Sandbox task The Sandbox test aims to evaluate behaviour of our system with Ã small scales. Table 3 below presents the results obtained by running EXONA on the instance matching track of OAEI campaign 2015 done with the Sandbox task.

	Precision	**Recall**	**F-measure**
EXONA	0.518	0.518	0.518
InsMT+	0.556	0.059	0.106
Lily	1.0	1.0	1.0
LogMap	1.0	1.0	1.0
RiMOM	1.0	1.0	1.0

Table 3. Results Author Recognition Task for Sandbox task

Mainbox task This task is also done with Mainbox task. The goal of this test is to evaluate the behaviour of our system in large scale. Table 4 below presents the results obtained by running EXONA on the instance matching track of OAEI campaign 2015 done with the Mainbox task.

	Precision	**Recall**	**F-measure**
EXONA	0.409	0.409	0.409
InsMT+	0.246	0.028	0.05
Lily	0.999	0.998	0.999
LogMap	0.999	1.0	0.999
RiMOM	0.999	0.999	0.999

Table 4. Results Author Recognition Task for Mainbox task

3 Conclusion

Exona participation in OAEI 2015 is encouraging although its participation is restricted to a few sub-cases, as it supplies good metric values in the two considered cases. Results reflects some strengths and some good aspects that need to be improved.

References

1. Euzenat, J., Shvaiko, P.: Ontology Matching (Second Edition). Springer-Verlag, Heidelberg (DE) (2013)

GMap: Results for OAEI 2015

Weizhuo Li and Qilin Sun

Institute of Mathematics,Academy of Mathematics and Systems Science,
Chinese Academy of Sciences, Beijing, P. R. China
{liweizhuo,sunqilin}@amss.ac.cn

Abstract. GMap is an alternative probabilistic scheme for ontology matching, which combines the sum-product network and the noisy-or model. More precisely, we employ the sum-product network to encode the similarities based on individuals and disjointness axioms. The noisy-or model is utilized to encode the probabilistic matching rules, which describe the influences among entity pairs across ontologies. In this paper, we briefly introduce GMap and its results of four tracks (i.e.,Benchmark, Conference, Anatomy and Ontology Alignment for Query Answering) on OAEI 2015.

1 Presentation of the system

1.1 State, purpose, general statement

The state of the art approaches have utilized probabilistic graphical models [5] for ontology matching such as OMEN [7], iMatch [1] and CODI [8]. However, few of them can keep inference tractable and ensure no loss in inference accuracy. In this paper, we propose an alternative probabilistic scheme, called GMap, combining the sum-product network (SPN) and the noisy-or model [6]. Except for the tractable inference, these two graphical models have some inherent advantages for ontology matching. For SPN, even if the knowledge such as individuals or disjointness axioms is missing, SPN can also calculate their contributions by the maximum a posterior (MAP) inference. For the noisy-or model, it is a reasonable approximation for incorporating probabilistic matching rules to describe the influences among entity pairs.

Figure 1 shows the sketch of GMap. Given two ontologies O_1 and O_2, we calculate the lexical similarity based on edit-distance, external lexicons and TFIDF [3] with the max strategy. Then, we employ SPN to encode the similarities based on individuals and disjointness axioms and calculate the contribution through MAP inference. After that, we utilize the noisy-or model to encode the probabilistic matching rules and the value calculated by SPN. With one-to-one constraint and crisscross strategy in the refine module, GMap obtains initial matches. The whole matching procedure is iterative. If there is no additional matches identified, the matching is terminated.

1.2 Specific techniques used

The similarities based on individuals and disjointness axioms In open world assumption, individuals or disjointness axioms are missing at times. Therefore, we define

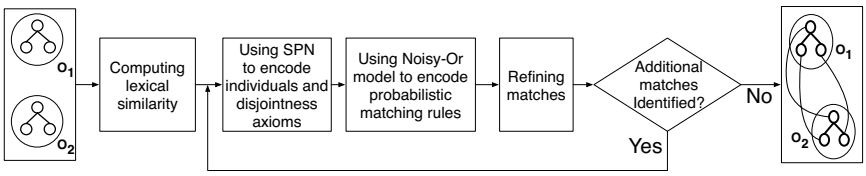


Fig. 1: Matching process in GMap

a special assignment—"$Unknown$" of the similarities based on these individuals and disjointness axioms.

For individuals, we employ the string equivalent to judge the equality of them. When we calculate the similarity of concepts based on individuals across ontologies, we regard individuals of each concept as a set and use Ochiai coefficient[1] to measure the value. We use a boundary t to divide the value into three assignments (i.e., 1, 0 and $Unknown$). Assignment 1 (or 0) means that the pair matches (or mismatches). If the value ranges between 0 and t or the individuals of one concept are missing, the assignment is $Unknown$.

For disjointness axioms, we utilize these axioms and subsumption relations within ontologies and define some rules to determine assignments of similarity. For example, x_1, y_1 and x_2 are concepts that come from O_1 and O_2. If x_1 matches x_2 and x_1 is disjoint with y_1, then y_1 is disjoint with x_2 as well as their descendants. The similarity also have three assignments. Assignment 1 (or 0) means the pair mismatches (or overlaps). If all the rules are not satisfied, the assignment is $Unknown$.

Using SPN to encode the simialrities based on individuals and disjointness axioms Sum-Product Network is a directed acyclic graph with weighted edges, where variables are leaves and internal nodes are sums and products [9]. As shown in Figure 2, we designed a sum-product network S to encode above similarities and calculate the contributions. All the leaves, called indicators, are binary-value. M represents the contribution of individuals and disjointness axioms and indicators M_1, M_2, M_3 comprise the assignments of it. $M_1 = 1$ (or $M_2 = 1$) means that the contribution is positive (or negative). If $M_3 = 1$, the contribution is $Unknown$. Similarly, Indicators D_0, D_1, I_1, I_2, I_3 correspond to assignments of the similarities based on individuals and disjointness axioms. The concrete assignment metrics are listed in Table 1–2 and the assignment metric of M is similar to the metric of similarity D.

Table 1: Metric for Similarity D

Assignments	Indicators
$D = 1$	$D_0 = 0, D_1 = 1$
$D = 0$	$D_0 = 1, D_1 = 0$
$D = Unknown$	$D_0 = 1, D_1 = 1$

Table 2: Metric for Similarity I

Assignments	Indicators
$I = 1$	$I_1 = 1, I_2 = 0, I_3 = 0$
$I = 0$	$I_1 = 0, I_2 = 1, I_3 = 0$
$I = Unknown$	$I_1 = 0, I_2 = 0, I_3 = 1$

[1] https://en.wikipedia.org/wiki/Cosine_similarity

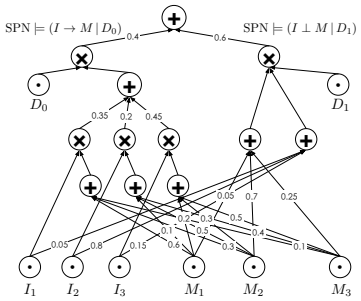


Fig. 2: The designed SPN : When $D_0 = 1, D_1 = 0$, it means that the distribution of M depends on the distribution of I; When $D_0 = 0, D_1 = 1$, the distributions of M and I are independent.

With the MAP inference in SPN [9], we can obtain the indicators' value of contribution M. The MAP inference has three steps. Firstly, replace sum nodes with max nodes. Secondly, with the bottom-up method, each max node can get a maximum weighted value. Finally, the downward pass starts from the root node and recursively selects the highest-value child of each max node, then the indicators' value of M are obtained. Moreover, even if individuals or disjointness axioms are missing at times, We can also calculate the contribution M by MAP inference. Assumed $I = 1$, $D = Unknown$ for one pair, then we can obtain $I_1 = 1, I_2 = 0, I_3 = 0, D_0 = 1, D_1 = 1$ with defined similarities and assignment metrics of SPN. As contribution M is not given, so we need to set $M_1 = 1, M_2 = 1, M_3 = 1$. After MAP inference, we observe $M_1 = 1$ which means that the contribution is positive. Moreover, it is able to infer $D_0 = 1$, which means the pair overlaps.

As the network S is complete and decomposable, the inference in S can be computed in time linear in the number of edges [4]. So MAP inference is tractable.

Combining the lexical similarity and the contribution calculated by SPN Considering the range of lexical similarity, we define a scaling factor α to limit the contribution of lexical similarity. It can help us to analyze the sources from different contributions. The SPN-based similarity (S_0) is defined in Eqs 1, which is calculated according to the indicators' value of M and D.

$$S_0(x_1, x_2) = \begin{cases} 0 & M_2 = 1, D_1 = 1 \\ \alpha * lexSim(x_1, x_2) + \lambda & M_1 = 1, D_0 = 1 \\ \alpha * lexSim(x_1, x_2) - \lambda & M_2 = 1, D_0 = 1 \\ \alpha * lexSim(x_1, x_2) & M_3 = 1, D_0 = 1 \end{cases} \tag{1}$$

where λ is a contribution factor that represents the contribution based on disjointness axioms and individuals. If contribution is positive (negative) and pair overlaps, the SPN-based similarity is equal to the scaled lexical similarity adding (subtracting) λ. If the contribution is $Unknown$ and pair overlaps, the SPN-based similarity is equal to the s-

caled lexical similarity. If the pair mismatches, then the inferred contribution is negative and the SPN-based similarity is equal to 0.

Using Noisy-Or model to encode probabilistic matching rules As listed in Table 3, we utilize probabilistic matching rules to describe the influences among the related pairs across ontologies.

Table 3: The probabilistic matching rules between entity pairs

ID	Category	Probabilistic matching rules
R_1	Class	two classes probably match if their fathers match
R_2	Class	two classes probably match if their children match
R_3	Class	two classes probably match if their siblings match
R_4	Class	two classes about domain probably match if related objectproperties match and range of these property match
R_5	Class	two classes about range probably match if related objectproperties match and domain of these properties match
R_6	Class	two classes about domain probably match if related dataproperties match and value of these properties match

Considering the matching probability of one pair, we observe that the condition of each rule has two value (i.e., T or F) and all the matching rules are independent of each other approximately. Moreover, all of them benefit to improving the matching probability of this pair. Therefore, we utilize the noisy-or model [5] to encode them.

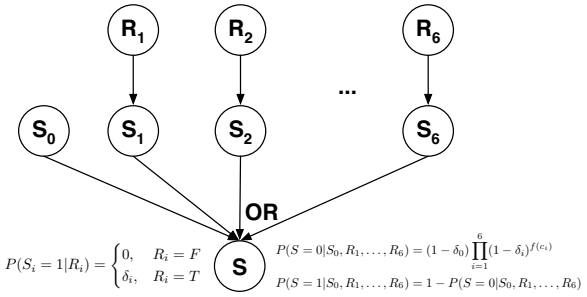


Fig. 3: The network structure of noisy-or model designed in GMap

Figure 3 shows the designed noisy-or model applied in concept pairs and the extension to property pairs is straight-forward, where R_i corresponds to the ith rule and S_i is the conditional probability depended on the condition of R_i. S_0 represents the SPN-based similarity which is a leak probability [5]. We can easily calculate the matching probability of each pair, $P(S = 1|S_0, R_1, ..., R_6)$, according to the formulas listed in this figure, where c_i is the count of satisfied R_i and sigmoid function $f(c_i)$ is used to limit the upper bound of contribution of R_i.

As the inference in the noisy-or model can be computed in time linear in size of nodes [5], so GMap can keep inference tractable in the whole matching process.

1.3 Adaptations made for the evaluation

There are two kinds of parameters that need be set. one mainly comes from networks and it is set manually based on some considerations [2]. The others are adapted by I3CON data set[2] such as scaling factor (α), contribution factor (λ) in Eqs 1 and threshold (θ). Nevertheless, we do not make any specific adaptation for OAEI 2015 evaluation campaign and all parameters are the same for different tracks.

1.4 Link to the system and parameters file

The latest version of GMap can be seen on https://github.com/liweizhuo001/GMap1.1.

1.5 Link to the set of provided alignments

The results of GMap can be seen on https://github.com/liweizhuo001/GMap1.1.

2 Results

In this section, we present the results of GMap achieved on OAEI 2015. Our system mainly focuses on Benchmark, Anatomy, Conference. Adding to that, we also present the results of the test Ontology Alignment for Query Answering which not follow the classical ontology alignment evaluation on the SEALS platform.

2.1 Benchmark

The goal of Benchmark is to evaluate the matching systems in scenarios where the input ontologies lack important information. Table 4 summarizes the average results[3] of it.

Table 4: Results for Benchmark track

Test	Precision	Recall	F-Measure
biblio	0.93	0.53	0.68
energy	0.32	0.02	0.11

GMap had a good performance in biblio, ranking third in F-measure, because it makes use of the string resource such as identifiers, labels and comments. Specially in ontologies 201–210 of biblio, as the mapping concepts have the same group of individuals but different names, SPN can play a role in improving the alignment quality of GMap.

[2] http://www.atl.external.lmco.com/projects/ontology/i3con.html
[3] The new test set about energy exists some troubles.

2.2 Anatomy

The Anatomy track consists of finding an alignment between the Adult Mouse Anatomy (2744 classes) and a part of the NCI Thesaurus (3304 classes) describing the human anatomy. The results are shown in Table 5.

Table 5: Results for Anatomy track

Matcher	Runtime (s)	Size	Precision	F-Measure	Recall	Recall+	Coherent
AML	40	1477	0.956	0.944	0.931	0.82	√
XMAP	50	1414	0.928	0.896	0.865	0.647	√
LogMapBio	895	1549	0.882	0.891	0.901	0.738	√
LogMap	24	1397	0.918	0.88	0.846	0.593	√
GMap	2362	1344	0.916	0.861	0.812	0.534	-

GMap ranked fifth in Anatomy track. We analyze that GMap does not concentrate on language techniques such as the abbreviations and emphasizes one-to-one constraint. Both of them may cause a low recall. In addition, these top-ranked systems employ alignment debugging techniques, which is helpful to improve alignment quality. However, we do not employ these techniques in the current version.

2.3 Conference

Conference track contains sixteen ontologies from the conference organization domain. There are two versions of reference alignment. The original reference alignment is labeled as RA1, and the new reference alignment, generated as a transitive closure computed on the original reference alignment, is labeled as RA2. Table 6 shows the results of our system in this track.

Table 6: Results for Conference track

	Precision	Recall	F-Measure
RA1	0.66	0.65	0.65
RA2	0.63	0.59	0.61

For Conference track, GMap ranked sixth of the 14 participants, which outperforms others in recall except AML but its precision is lower than them. There are mainly two reasons. One is the lexical similarity which combines the similarities based on edit-distance, external lexicons and TFIDF with the max strategy. The other is the noisy-or model which is hard to describe the negative effect on pairs matching [5]. Both of them would retain some false positive matches after matching finished. Specially in property pairs, even though their domains and ranges mismatch, GMap can not describe this negative impact. Therefore, employing alignment debugging techniques are comparatively ideal method solutions to deal with this problem.

2.4 Ontology Alignment for Query Answering (OA4QA)

The aims of OA4QA are investigating the effects of logical violations affecting computed alignments and evaluating the effectiveness of repair strategies employed by the matchers. In the OAEI 2015 the ontologies and reference alignment (RA1) are based on the conference track. RAR1 is a repaired version of RA1 different from RA2 in the conference track. The table 7 presents the results for the whole set of queries.

Table 7: Results for OA4QA track

Matcher	Answered queries	RA1			RAR1		
		P	R	F	P	R	F
GMap	9/18	0.324	0.389	0.343	0.303	0.389	0.330

Since GMap did not consider mapping repair techniques, it was only able to answer half of queries, which influenced the obtained precision and recall at last.

3 General comments

3.1 Comments on the results

GMap achieved qualified results in its first participation in OAEI, which is competitive with other systems in some tracks such as Benchmark, Conference, Anatomy. Both of the employed graphical models are able to improve the quality of alignment in terms of the defined lexical similarity [6]. Most improvements are attributed to the noisy-or model because it makes use of rich relations specified in ontologies such as in Anatomy track. If there are some individuals and disjointness axioms declared in ontologies, SPN will work such as biblio (201–210) in Benchmark track. More importantly, Combining SPN and the noisy-or model is able to increase precision and recall further.

However, some weaknesses still remain. For example, the alignment incoherence of GMap is unsolved, which influences the performance of GMap. In addition, it is important for us to consider the efficiency of GMap such as running time and memory usage for large-scale mapping problems.

3.2 Discussions on the way to improve the proposed system

GMap still has a lot of room for improvement. Employing alignment debugging techniques are able to solve the alignment incoherent and reduce some false positive matches in alignment such as the pair {Conference: has_members, edas: hasMember} in Conference track. In addition, seeking available data sets to learn parameters of the sum-product network and the noisy-or model is also one direction of our future works.

4 Conclusion

In this paper, we have presented GMap and its results of four tracks (i.e.,Benchmark, Conference, Anatomy and Ontology Alignment for Query Answering) on OAEI 2015. The results show that GMap is competitive with the top-ranked systems in some tracks by means of combining some special graphical models (i.e.,SPN, Noisy-or model). On the other hand, for those disadvantages exposed, we discuss the possible solutions. In the future, we would like to participate in more tracks and hope to efficiently solve the instance matching and large biomedical ontologies matching challenges.

Acknowledgments. This research was partly supported by the Natural Science Foundation of China (No. 61232015), the National Key Research and Development Program of China (Grant No. 2002CB312004), the Knowledge Innovation Program of the Chinese Academy of Sciences, Key Lab of Management, Decision and Information Systems of CAS, Institute of Computing Technology of CAS, and the Key Laboratory of Multimedia and Intelligent Software at Beijing University of Technology.

References

1. Albagli, S., Ben-Eliyahu-Zohary, R., Shimony, S.E.: Markov network based ontology matching. Journal of Computer and System Sciences **78**(1) (2012) 105–118
2. Ding, L., Finin, T.: Characterizing the semantic web on the web. In: The Semantic Web-ISWC 2006. Springer (2006) 242–257
3. Euzenat, J., Shvaiko, P.: Ontology Matching. Springer Science & Business Media (2013)
4. Gens, R., Pedro, D.: Learning the structure of sum-product networks. In: Proceedings of The 30th International Conference on Machine Learning. (2013) 873–880
5. Koller, D., Friedman, N.: Probabilistic graphical models: principles and techniques. MIT press (2009)
6. Li, W.: Combining sum-product network and noisy-or model for ontology matching
7. Mitra, P., Noy, N.F., Jaiswal, A.R.: Omen: A probabilistic ontology mapping tool. In: The Semantic Web–ISWC 2005. Springer (2005) 537–547
8. Niepert, M., Meilicke, C., Stuckenschmidt, H.: A probabilistic-logical framework for ontology matching. In: AAAI, Citeseer (2010)
9. Poon, H., Domingos, P.: Sum-product networks: A new deep architecture. In: Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on, IEEE (2011) 689–690

InsMT+ Results for OAEI 2015 Instance Matching

Abderrahmane Khiat[1] , Moussa Benaissa[1]

LITIO Laboratory, University of Oran1 Ahmed Ben Bella, Oran, Algeria
abderrahmane_khiat@yahoo.com , moussabenaissa@yahoo.fr

Abstract. The InsMT+ is an improved version of InsMT system participated at OAEI 2014. The InsMT+ an automatic instance matching system which consists in identifying the instances that describe the same real-world objects. The InsMT+ applies different string-based matchers with a local filter. This is the second participation of our system and we have improved somehow the results obtained by the previous version.

Keywords: Terminological Techniques, String Based Similarity, Instance Mapping, Instance Matching, Linked Data, Web of Data, Semantic Interoperability, Semantic Web.

1 Presentation of the System

1.1 State, Purpose, General Statement

The *objective* of *Linked Data* with the emergence of *the Web of Data* is to *interlink semantically data together* in order to be *reused and processed automatically* by the *software agents*. These *data* described by *instances* are *heterogeneous* and *distributed*. The *Instance matching* is a very necessary task in Linked Data; it aims to identify the *instances* that *describe* the *same real-world objects*.

The *enormous volume* of data already available on the web and its continuity to increase, requires techniques and tools capable to identify the instances that describe the same real-world objects automatically.

In this paper, we describe InsMT+ an improved version of our InsMT system which participated in OAEI 2014. This second version consists to apply *different string-based matchers* with *a local filter*. The second version shows good results better than the previous one but still not very satisfiable. The details of each step of our system are described in the following section.

1.2 Specific Techniques Used

The process of our system consists in the following successive steps.

Step 1: Extraction and Normalization of Instances In this step, our system extracts the instances. Then, we have applied (1) case conversion (conversion of all words in same upper or lower case) and (2) stop word elimination to normalize the instance informations.

Step 2: Terminological Matchers In this step, our system calculates the similarities between instances, normalized in previous phase, using various string-based matching algorithms. More precisely the different string-based matching algorithms used are: levenshtein-distance, Jaro, SLIM-Winkler. The calculations of similarities by each string matching algorithm are represented in matrix.

Step 3: Local Filter In this step, our system applies a local filter on each matrix i.e. we choose for each string-based matching algorithm a threshold to realize a filter. We consider that: the similarities which are less than the threshold are set to 0. Our intuition behind this local filter is that the similarities which are less than the threshold can influence the strategy of the average aggregation.

Step 4: Aggregation of Similarities In this step, our system combines the similarities of each matrix (after we have applied a local filter) using the average aggregation method and the result of the aggregation is represented in a matrix.

Step 5: Global Filter and Identification of Alignment In this step, our system applies a second filter on the combined matrix (result of the previous step) in order to select the correspondences found using the maximum strategy with a threshold.

1.3 Adaptations Made for the Evaluation

We do not have made any specific adaptation for this first version of InsMT+, for OAEI 2015 evaluation campaign. All parameters are the same for instance matching track of OAEI 2015.

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

The result of InsMT+ system can be downloaded from OAEI 2015 website `http://islab.di.unimi.it/im_oaei_2015/index.html`

2 Results

In this section, we present the results obtained by running InsMT+ on instance matching track of OAEI 2015 evaluation campaign.

2.1 Author Disambiguation Task

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

We present below the results obtained by running InsMT+ system on author disambiguation task (see Tab. 1).

Table 1: The results of InsMT+ on the Author Disambiguation Task of OAEI 2015.

Track	System	Expected mappings	Retrieved mappings	Precision	Recall	F-measure
Sandbox task	EXONA	854	854	0.941	0.941	0.941
Mainbox task	EXONA	8428	144827	0.0	0.0	NaN
Sandbox task	InsMT+	854	722	0.834	0.705	0.764
Mainbox task	InsMT+	8428	7372	0.76	0.665	0.709
Sandbox task	Lily	854	854	0.981	0.981	0.981
Mainbox task	Lily	8428	8428	0.964	0.964	0.964
Sandbox task	LogMap	854	779	0.994	0.906	0.948
Mainbox task	LogMap	8428	7030	0.996	0.831	0.906
Sandbox task	RiMOM	854	854	0.929	0.929	0.929
Mainbox task	RiMOM	8428	8428	0.911	0.911	0.911

* The results of InsMT+ are better compared to the first version participated in OAEI 2014, we can say that we have improved the results in terms of precision. However, the results are less better than other systems due to the simple techniques used in InsMT+. Since, InsMT+ is based only on String-based similarity.

2.2 Author Recognition Task

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.

We present below the results obtained by running InsMT+ system on author recognition task (see Tab. 2).

Table 2: The results of InsMT+ on the Author Recognition Task of OAEI 2015.

Track	System	Expected mappings	Retrieved mappings	Precision	Recall	F-measure
Sandbox task	EXONA	854	854	0.518	0.518	0.518
Mainbox task	EXONA	8428	8428	0.409	0.409	0.409
Sandbox task	InsMT+	854	90	0.556	0.059	0.106
Mainbox task	InsMT+	8428	961	0.246	0.028	0.05
Sandbox task	Lily	854	854	1.0	1.0	1.0
Mainbox task	Lily	8428	8424	0.999	0.998	0.999
Sandbox task	LogMap	854	854	1.0	1.0	1.0
Mainbox task	LogMap	8428	8436	0.999	1.0	0.999
Sandbox task	RiMOM	854	854	1.0	1.0	1.0
Mainbox task	RiMOM	8428	8428	0.999	0.999	0.999

* The results of InsMT+ on this track are not at all very satisfiable. However, we can remark that the number of retrieved mappings by our system is less 10 time than the mappings discovered by other systems, which explained the results obtained. We are trying to analyses the reason of these results in order to improve our system.

3 Conclusion

This is the second time that InsMT+ system has participated in SEAL platform and OAEI campaign. In this year, our system has participated only in two instance matching tracks of OAEI 2015 evaluation campaign. The InsMT+ system gives good results better than the InsMT system but these results still not statifaisable. As future Perspective, we attempt to improve more our system in order to get better results.

References

1. A. Doan, J. Madhavan, P. Domingos, and A. Halevy, Learning to map ontologies on the semantic web, in Proceedings of the International World Wide Web Conference (2003).
2. A. Maedche and V. Zacharias, Clustering ontologybased metadata in the semantic web, in Proceedings of the 13th ECML and 6th PKDD, (2002).
3. A. Khiat, M. Benaissa, InsMT / InsMTL results for OAEI 2014 instance matching. In Proceedings of the 9th International Workshop on Ontology Matching co-located with the 13th International Semantic Web Conference (ISWC 2014), October 20, pp. 120-125. CEURWS.org, Trentino, Italy, 2014.
4. A. Maedche, B. Motik, N. Silva and R. Volz "Mafraa mappingframework for distributed ontologies", Springer, Benjamins VR (eds) EKAW, Berlin, vol 2473, pp 235250, (2002).
5. K. Todorov, P. Geibel, KU. Kuhnberger "Mining concept similarities for heterogeneous ontologies", Springer, Berlin, ICDM, vol 6171. , pp 86100, (2010).
6. J. Euzenat and P. Valtchev, Similarity-based ontology alignment in owllite, in Proceedings of ECAI, (2004).
7. J. Euzenat and P. Shvaiko. OntologyMatching. Springer (2007).
8. M. Ehrig. Ontology Alignment Bridging the Semantic Gap. Springer (2007).
9. M. Jaro. Advances in record-linkage methodology as applied to matching the 1985 census of tampa, florida. Journal of America Statistical Association, 84(406):414-420, (1989).
10. A. Khiat et M. Benaissa: "Nouvelle Approche dAlignement dOntologies base dInstances : trasferet des instances par linfrence", In The Proceeding of International Conference On Artificial Intelligence and Information Technology, ICA2IT 2014, Ouargla, Algeria, (2014).
11. V. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. Soviet Physics Doklady, 10:707-710, (1966).
12. W. Winkler. The state of record linkage and current research problems. Statistics of Income Division, Internal Revenue Service. Publication R99/04 (1999).
13. M. Ehrig and Y. Sure, Ontology mapping - an integrated approach, in Proceedings of the European Semantic Web Symposium ESWS, (2004).
14. B. Schopman, S. Wang, A. Isaac and S. Schlobach, Instance-Based Ontology Alignment by Instance Enrichment, Journal on Data Semantics, vol. 1, N 4, (2012).
15. E. Rahm Towards large-scale schema and ontology Alignment, ReCALL, (2011).
16. J. Li, J. Tang, Y. Li and Q. Luo, Rimom: a dynamic multistrategy ontology alignment framework, IEEE Trans Knowl, (2009).

Lily Results for OAEI 2015

Wenyu Wang[1,2], Peng Wang[1]

[1] School of Computer Science and Engineering, Southeast University, China
[2] Chien-Shiung Wu College, Southeast University, China
{ms, pwang} @ seu.edu.cn

Abstract. This paper presents the results of Lily in the ontology alignment contest OAEI 2015. As a comprehensive ontology matching system, Lily is intended to participate in four tracks of the contest: benchmark, conference, anatomy, and instance matching. The specific techniques used by Lily will be introduced briefly. The strengths and weaknesses of Lily will also be discussed.

1 Presentation of the system

With the use of hybrid matching strategies, Lily, as an ontology matching system, is capable of solving some issues related to heterogeneous ontologies. It can process normal ontologies, weak informative ontologies [5], ontology mapping debugging [7], and ontology matching tunning [9], in both normal and large scales. In previous OAEI contests [1–3], Lily has achieved preferable performances in some tasks, which indicated its effectiveness and wideness of availability.

1.1 State, purpose, general statement

The core principle of matching strategies of Lily is utilizing the useful information correctly and effectively. Lily combines several effective and efficient matching techniques to facilitate alignments. There are five main matching strategies: (1) Generic Ontology Matching (GOM) is used for common matching tasks with normal size ontologies. (2) Large scale Ontology Matching (LOM) is used for the matching tasks with large size ontologies. (3) Instance Ontology Matching (IOM) is used for instance matching tasks. (4) Ontology mapping debugging is used to verify and improve the alignment results. (5) Ontology matching tuning is used to enhance overall performance.

The matching process mainly contains three steps: (1) Pre-processing, when Lily parses ontologies and prepares the necessary information for subsequent steps. Meanwhile, the ontologies will be generally analyzed, whose characteristics, along with studied datasets, will be utilized to determine parameters and strategies. (2) Similarity computing, when Lily uses special methods to calculate the similarities between elements from different ontologies. (3) Post-processing, when alignments are extracted and refined by mapping debugging.

In this year, some algorithms and matching strategies of Lily have been modified for higher efficiency, and adjusted for brand-new matching tasks like Author Recognition and Author Disambiguation in the Instance Matching track.

1.2 Specific techniques used

Lily aims to provide high quality 1:1 concept pair or property pair alignments. The main specific techniques used by Lily are as follows.

Semantic subgraph An element may have heterogeneous semantic interpretations in different ontologies. Therefore, understanding the real local meanings of elements is very useful for similarity computation, which are the foundations for many applications including ontology matching. Therefore, before similarity computation, Lily first describes the meaning for each entity accurately. However, since different ontologies have different preferences to describe their elements, obtaining the semantic context of an element is an open problem. The semantic subgraph was proposed to capture the real meanings of ontology elements [4]. To extract the semantic subgraphs, a hybrid ontology graph is used to represent the semantic relations between elements. An extracting algorithm based on an electrical circuit model is then used with new conductivity calculation rules to improve the quality of the semantic subgraphs. It has been shown that the semantic subgraphs can properly capture the local meanings of elements [4].

Based on the extracted semantic subgraphs, more credible matching clues can be discovered, which help reduce the negative effects of the matching uncertainty.

Generic ontology matching method The similarity computation is based on the semantic subgraphs, which means all the information used in the similarity computation comes from the semantic subgraphs. Lily combines the text matching and structure matching techniques.

Semantic Description Document (SDD) matcher measures the literal similarity between ontologies. A semantic description document of a concept contains the information about class hierarchies, related properties and instances. A semantic description document of a property contains the information about hierarchies, domains, ranges, restrictions and related instances. For the descriptions from different entities, the similarities of the corresponding parts will be calculated. Finally, all separated similarities will be combined with the experiential weights.

Matching weak informative ontologies Most existing ontology matching methods are based on the linguistic information. However, some ontologies may lack in regular linguistic information such as natural words and comments. Consequently the linguistic-based methods will not work. Structure-based methods are more practical for such situations. Similarity propagation is a feasible idea to realize the structure-based matching. But traditional propagation strategies do not take into consideration the ontology features and will be faced with effectiveness and performance problems. Having analyzed the classical similarity propagation algorithm, *Similarity Flood*, we proposed a new structure-based ontology matching method [5]. This method has two features: (1) It has more strict

but reasonable propagation conditions which lead to more efficient matching processes and better alignments. (2) A series of propagation strategies are used to improve the matching quality. We have demonstrated that this method performs well on the OAEI benchmark dataset [5].

However, the similarity propagation is not always perfect. When more alignments are discovered, more incorrect alignments would also be introduced by the similarity propagation. So Lily also uses a strategy to determine when to use the similarity propagation.

Large scale ontology matching Matching large ontologies is a challenge due to its significant time complexity. We proposed a new matching method for large ontologies based on reduction anchors [6]. This method has a distinct advantage over the divide-and-conquer methods because it does not need to partition large ontologies. In particular, two kinds of reduction anchors, positive and negative reduction anchors, are proposed to reduce the time complexity in matching. Positive reduction anchors use the concept hierarchy to predict the ignorable similarity calculations. Negative reduction anchors use the locality of matching to predict the ignorable similarity calculations. Our experimental results on the real world datasets show that the proposed methods are efficient in matching large ontologies [6].

Ontology mapping debugging Lily utilizes a technique named *ontology mapping debugging* to improve the alignment results [7]. Different from existing methods that focus on finding efficient and effective solutions for the ontology mapping problems, mapping debugging emphasizes on analyzing the mapping results to detect or diagnose the mapping defects. During debugging, some types of mapping errors, such as redundant and inconsistent mappings, can be detected. Some warnings, including imprecise mappings or abnormal mappings, are also locked by analyzing the features of mapping result. More importantly, some errors and warnings can be repaired automatically or can be presented to users with revising suggestions.

Ontology matching tuning Lily adopted ontology matching tuning this year. By performing parameter optimization on training datasets [9], Lily is able to determine the best parameters for similar tasks. Those data will be stored. When it comes to real matching tasks, Lily will perform statistical calculations on the new ontologies to acquire their features that help it find the most suitable configurations, based on previous training data. In this way, the overall performance can be improved.

Currently, ontology matching tuning is not totally automatic. It is difficult to find out typical statistical parameters that distinguish each task from others. Meanwhile, learning from test datasets can be really time-consuming. Our experiment is just a beginning.

1.3 Adaptations made for the evaluation

For benchmark, anatomy and conference tasks, Lily is totally automatic, which means Lily can be invoked directly from the SEALS client. It will also determine which strategy to use and the corresponding parameters. For a specific instance matching task, Lily needs to be configured and started up manually, so only matching results were submitted.

1.4 Link to the system and parameters file

SEALS wrapped version of Lily for OAEI 2015 is available at `https://drive.google.com/file/d/0B4fqkE38d3QrS1Zta0pPSFpqXzA/view?usp=sharing`.

1.5 Link to the set of provided alignments

The set of provided alignments, as well as overall performance, is available at each track of the OAEI 2015 official website, `http://oaei.ontologymatching.org/2015/`.

2 Results

2.1 Benchmark track

There are two datasets in different sizes: *Biblio* and *energy*. The former one, which will be matched using Generic Ontology Matching, is generally small, while the latter one is so much that it has to be matched by Large scale Ontology Matching.

There are five groups of test suites in each dataset. Each test suite has 94 matching tasks. The overall results of one test suite will be represented by the mean value of Precision, Recall and F-Measure. Test suites were generated from the same seed ontologies, which means they are all equal. Thus, the harmonic mean values of all test suites will be used to evaluate how well Lily worked.

The detailed results are shown in Table 1.

Table 1. The performance in the Benchmark track

Test suite	Precision	Recall	F-Measure
biblio-r1	0.96	0.83	0.89
biblio-r2	0.96	0.83	0.89
biblio-r3	0.97	0.84	0.90
biblio-r4	0.97	0.83	0.89
biblio-r5	0.97	0.84	0.90
H-mean	**0.97**	**0.83**	**0.90**
energy-r1	0.90	0.76	0.82
energy-r2	0.90	0.77	0.83
energy-r3	0.90	0.77	0.83
energy-r4	0.89	0.76	0.82
energy-r5	0.91	0.77	0.83
H-mean	**0.90**	**0.77**	**0.83**

As Table 1 has shown, Lily handles Benchmark datasets well in both small and large scales, although the results of the *energy* dataset are slightly worse as the expense of better performance. According to the Benchmark results of OAEI2015[1], Lily has the highest overall F-Measure among 11 matching systems that generated alignments for the *Biblio* dataset. However, the public results show that Lily failed to produce alignments for *energy* dataset. That is because the *energy* dataset is a replacement for its former dataset *IFC*. The substitution also brought about format changes of ontology description files. Consequently, Lily and some other systems were not able to parse ontologies correctly. After the issue was fixed, we evaluated Lily on only *energy* dataset with SEALS client and obtained the results.

2.2 Anatomy track

The anatomy matching task consists of two real large-scale biological ontologies. Table 2 shows the performance of Lily in the Anatomy track on a server with one 3.46 GHz, 6-core CPU and 8GB RAM allocated. The time unit is second (s).

Table 2. The performance in the Anatomy track

Matcher	Runtime	Precision	Recall	F-Measure
Lily	266s	0.87	0.79	0.83

Compared with the result in OAEI 2011 [8], there is a small improvement of Precision, Recall and F-Measure, from 0.80, 0.72 and 0.76 to 0.87, 0.79 and 0.83,

[1] `http://oaei.ontologymatching.org/2015/results/benchmarks/index.html`

respectively. One main reason for the improvement is that we found the names of classes not semantically useful, which would confuse Lily when the similarity matrix was calculated. After the names were excluded, better alignments were generated. Besides, there is a significant reduction of the time consumption, from 563s to 266s. This is not only the result of stronger CPU, but also because more optimizations, like parallelization, were applied to the algorithms in Lily.

However, as can be seen in the overall result, Lily lies in the middle position of the rank, which indicates it is still possible to make further progress. Additionally, some key algorithms have not been successfully parallelized. After that is done, the time consumption is expected to be further reduced.

2.3 Conference track

In this track, there are 7 independent ontologies that can be matched with one another. The 21 subtasks are based on given reference alignments. As a result of heterogeneous characters, it is a challenge to generate high-quality alignments for all ontology pairs in this track.

Lily adopted ontology matching tuning for the Conference track this year. Table 3 shows its latest performance.

Table 3. The performance in the Conference track

Test Case ID	Precision	Recall	F-Measure
cmt-conference	0.53	0.6	0.56
cmt-confof	0.80	0.25	0.38
cmt-edas	0.64	0.54	0.58
cmt-ekaw	0.55	0.55	0.55
cmt-iasted	0.57	1.00	0.73
cmt-sigkdd	0.70	0.58	0.64
conference-confof	0.67	0.53	0.59
conference-edas	0.41	0.41	0.41
conference-ekaw	0.62	0.64	0.63
conference-iasted	0.67	0.43	0.52
conference-sigkdd	0.71	0.67	0.69
confof-edas	0.69	0.47	0.56
confof-ekaw	0.79	0.75	0.77
confof-iasted	0.46	0.67	0.55
confof-sigkdd	0.17	0.14	0.15
edas-ekaw	0.67	0.52	0.59
edas-iasted	0.50	0.37	0.42
edas-sigkdd	0.63	0.33	0.43
ekaw-iasted	0.50	0.80	0.62
ekaw-sigkdd	0.50	0.46	0.48
iasted-sigkdd	0.56	0.67	0.61
Average	**0.59**	**0.53**	**0.56**

Compared with the result in OAEI 2011 [8], there is a significant improvement of mean Precision, Recall and F-Measure, from 0.36, 0.47 and 0.41 to 0.59, 0.53 and 0.56, respectively. Besides, all the tasks share the same configurations, so it is possible to generate better alignments by assigning the most suitable parameters for each task. We will continue to enhance this feature.

2.4 Instance matching track

We submitted alignments for two tasks in the IM track of OAEI 2015: Author Disambiguation Task and Author Recognition Task. For the other three tasks, there is currently no specific strategy available, so Lily will not produce alignments for them.

For each task, there are two matching subtasks with different scales. The *sandbox* scale is around 1,000 instances, which was provided as the test dataset. The *mainbox* scale is around 10,000 instances. The results will be analyzed for each task.

Author Disambiguation Task Lily utilized a different strategy for this task, as we found several features of the dataset: one author's name in ontology A usually contains the corresponding name in ontology B, and a slight difference of one property may distinguish publications in two ontologies. The result is shown in Table 4.

Table 4. The performance in the author-dis task

Matcher	Scale	Precision	Recall	F-Measure
Lily	sandbox	0.98	0.98	0.98
Lily	mainbox	0.96	0.96	0.96

As can be seen in Table 4, the strategy is practical. Most correct matches can be found with high precision in both *sandbox* and *mainbox* subtasks. According to overall results, Lily scores highest in this task. However, there are still some missing matches. After analyzing the reference alignments and matching ontologies, we found that some matched authors had actually no publication in common, and that accounts for many matches missed by Lily.

Author Recognition Task Quite different from the previous task, this task requires computations over the source ontology, whose results will be matched with the target ontology. Lily will first follow the requirement to generate an intermediate, statistical ontology from the source ontology. Then, string properties and numeric properties of that ontology and the target ontology will be compared in different methods. Finally, all the similarities will be combined. The result is shown in Table 5.

Table 5. The performance in the author-rec task

Matcher	Scale	Precision	Recall	F-Measure
Lily	sandbox	1.00	1.00	1.00
Lily	mainbox	0.99	0.99	0.99

As can be seen in Table 5, the strategy is practical as well, especially for the *sandbox* subtask.

3 General comments

In this year, a lot of modifications were done to Lily for both effectiveness and efficiency. The performance has been improved as we have expected. The strategies for new tasks have been proved to be useful.

On the whole, Lily is a comprehensive ontology matching system with the ability to handle multiple types of ontology matching tasks, of which the results are generally competitive. However, Lily still lacks in strategies for some newly developed matching tasks. The relatively high time and memory consumption also prevent Lily from finishing some challenging tasks.

4 Conclusion

In this paper, we briefly introduced our ontology matching system Lily. The matching process and the special techniques used by Lily were presented, and the alignment results were carefully analyzed.

There is still so much to do to make further progress. Lily needs more optimization to handle large ontologies with limited time and memory. Thus, techniques like parallelization will be applied more. Also, we have just tried out ontology matching tuning. With further research on that, Lily will not only produce better alignments for tracks it was intended for, but also be able to participate in the interactive track.

References

[1] Peng Wang, Baowen Xu: Lily: ontology alignment results for OAEI 2009. In The 4th International Workshop on Ontology Matching, Washington Dc., USA (2009)
[2] Peng Wang, Baowen Xu: Lily: Ontology Alignment Results for OAEI 2008. In The Third International Workshop on Ontology Matching, Karlsruhe, Germany (2008)
[3] Peng Wang, Baowen Xu: LILY: the results for the ontology alignment contest OAEI 2007. In The Second International Workshop on Ontology Matching (OM2007), Busan, Korea (2007)
[4] Peng Wang, Baowen Xu, Yuming Zhou: Extracting Semantic Subgraphs to Capture the Real Meanings of Ontology Elements. Journal of Tsinghua Science and Technology, vol. 15(6), pp. 724-733 (2010)

[5] Peng Wang, Baowen Xu: An Effective Similarity Propagation Model for Matching Ontologies without Sufficient or Regular Linguistic Information, In The 4th Asian Semantic Web Conference (ASWC2009), Shanghai, China (2009)
[6] Peng Wang, Yuming Zhou, Baowen Xu: Matching Large Ontologies Based on Reduction Anchors. In The Twenty-Second International Joint Conference on Artificial Intelligence (IJCAI 2011), Barcelona, Catalonia, Spain (2011)
[7] Peng Wang, Baowen Xu: Debugging Ontology Mapping: A Static Approach. Computing and Informatics, vol. 27(1), pp. 2136 (2008)
[8] Peng Wang: Lily results on SEALS platform for OAEI 2011. Proc. of 6th OM Workshop, pp. 156-162 (2011)
[9] Yang, Pan, Peng Wang, Li Ji, Xingyu Chen, Kai Huang, Bin Yu: Ontology Matching Tuning Based on Particle Swarm Optimization: Preliminary Results. In The Semantic Web and Web Science, pp. 146-155 (2014)

LogMap family results for OAEI 2015

E. Jiménez-Ruiz[1], B. Cuenca Grau[1], A. Solimando[2], and V. Cross[3]

[1] Department of Computer Science, University of Oxford, Oxford UK
[2] Inria Saclay and Université Paris-Sud, France
[3] Computer Science and Software Engineering, Miami University, Oxford, OH, United States

Abstract. We present the results obtained in the OAEI 2015 campaign by our ontology matching system LogMap and its variants: LogMapC, LogMapBio and LogMapLt. The LogMap project started in January 2011 with the objective of developing a scalable and logic-based ontology matching system. This is our sixth participation in the OAEI and the experience has so far been very positive. Currently, LogMap is the only system that participates in all OAEI tasks.

1 Presentation of the system

Ontology matching systems typically rely on lexical and structural heuristics and the integration of the input ontologies and the mappings may lead to many undesired logical consequences. In [12] three principles were proposed to minimize the number of potentially unintended consequences, namely: *(i) consistency principle*, the mappings should not lead to unsatisfiable classes in the integrated ontology; *(ii) locality principle*, the mappings should link entities that have similar *neighbourhoods*; *(iii) conservativity principle*, the mappings should not introduce alterations in the classification of the input ontologies. Violations to these principles may hinder the usefulness of ontology mappings. The practical effect of these violations, however, is clearly evident when ontology alignments are involved in complex tasks such as query answering [19].

LogMap [11, 13] is a highly scalable ontology matching system that implements the consistency and locality principles. LogMap also supports (real-time) user interaction during the matching process, which is essential for use cases requiring very accurate mappings. LogMap is one of the few ontology matching system that *(i)* can efficiently match semantically rich ontologies containing tens (and even hundreds) of thousands of classes, *(ii)* incorporates sophisticatedhttp://iswc2015.semanticweb.org/ reasoning and repair techniques to minimise the number of logical inconsistencies, and *(iii)* provides support for user intervention during the matching process.

LogMap relies on the following elements, which are keys to its favourable scalability behaviour (see [11, 13] for details).

Lexical indexation. An inverted index is used to store the lexical information contained in the input ontologies. This index is the key to efficiently computing an initial set of mappings of manageable size. Similar indexes have been successfully used in information retrieval and search engine technologies [2].

Logic-based module extraction. The practical feasibility of unsatisfiability detection and repair critically depends on the size of the input ontologies. To reduce the size of

the problem, we exploit ontology modularisation techniques. Ontology modules with well-understood semantic properties can be efficiently computed and are typically much smaller than the input ontology (e.g. [5]).

Propositional Horn reasoning. The relevant modules in the input ontologies together with (a subset of) the candidate mappings are encoded in LogMap using a Horn propositional representation. Furthermore, LogMap implements the classic Dowling-Gallier algorithm for propositional Horn satisfiability [6]. Such encoding, although incomplete, allows LogMap to detect unsatisfiable classes soundly and efficiently.

Axiom tracking. LogMap extends Dowling-Gallier's algorithm to track all mappings that may be involved in the unsatisfiability of a class. This extension is key to implementing a highly scalable repair algorithm.

Local repair. LogMap performs a greedy local repair; that is, it repairs unsatisfiabilities on-the-fly and only looks for the first available repair plan.

Semantic indexation. The Horn propositional representation of the ontology modules and the mappings is efficiently indexed using an interval labelling schema [1] — an optimised data structure for storing directed acyclic graphs (DAGs) that significantly reduces the cost of answering taxonomic queries [4, 20]. In particular, this semantic index allows us to answer many entailment queries as an index lookup operation over the input ontologies and the mappings computed thus far, and hence without the need for reasoning. The semantic index complements the use of the propositional encoding to detect and repair unsatisfiable classes.

1.1 LogMap variants in the 2015 campaign

As in the 2014, in the 2015 campaign we have participated with 3 variants:

LogMapLt is a "lightweight" variant of LogMap, which essentially only applies (efficient) string matching techniques.

LogMapC is a variant of LogMap which, in addition to the consistency and locality principles, also implements the conservativity principle (see details in [21, 22]). The repair algorithm is more aggressive than in LogMap, thus we expect highly precise mappings but with a significant decrease in recall.

LogMapBio includes an extension to use BioPortal [8, 9] as a (dynamic) provider of mediating ontologies instead of relying on a few preselected ontologies [3].

1.2 Adaptations made for the 2015 evaluation

LogMap's algorithm described in [11, 13, 14] has been adapted with the following new functionalities:

i **Local repair with global information.** We have extended LogMap to include global information in the local repairs, that is, repair plans of the same size are ordered according to their degree of conflictness (i.e. number of cases where the mappings in the repair are involved in an unsatisfiability). Hencee, LogMap prefers to remove mappings that are more likely to lead to other unsatisfiabilities.

ii **Extended multilingual support.** We have extended our multilingual module to use both *google translate* and *microsoft translator*.[4] Additionally, in order to split Chinese words, we rely on the ICTCLAS library[5] developed by the Institute of Computing Technology of the Chinese Academy of Sciences.

iii **Extended instance matching support.** We have also adapted LogMap's instance matching module to cope with the new OAEI 2014 tasks.

iv **BioPortal module.** In the OAEI 2015, LogMapBio uses the top-10 mediating (the 2014 version used only the top-5) ontologies given by the algorithm presented in [3]. Note that, LogMapBio only participates in the biomedical tracks. In the other tracks the results are expected to be the same as LogMap.

1.3 Link to the system and parameters file

LogMap is open-source and released under GNU Lesser General Public License 3.0.[6] LogMap components and source code are available from the LogMap's GitHub page: `https://github.com/ernestojimenezruiz/logmap-matcher/`.

LogMap distributions can be easily customized through a configuration file containing the matching parameters.

LogMap, including support for interactive ontology matching, can also be used directly through an AJAX-based Web interface: `http://csu6325.cs.ox.ac.uk/`. This interface has been very well received by the community since it was deployed in 2012. More than 2,000 requests coming from a broad range of users have been processed so far.

1.4 Modular support for mapping repair

Only a very few systems participating in the OAEI competition implement repair techniques. As a result, existing matching systems (even those that typically achieve very high precision scores) compute mappings that lead in many cases to a large number of unsatisfiable classes.

We believe that these systems could significantly improve their output if they were to implement repair techniques similar to those available in LogMap. Therefore, with the goal of providing a useful service to the community, we have made LogMap's ontology repair module (LogMap-Repair) available as a self-contained software component that can be seamlessly integrated in most existing ontology matching systems [16, 7].

2 Results

Please refer to `http://oaei.ontologymatching.org/2015/results/index.html` for the results of the LogMap family in the OAEI 2015 campaign.

[4] Currently we rely on the (unofficial) APIs available at `https://code.google.com/p/google-api-translate-java/` and `https://code.google.com/p/microsoft-translator-java-api/`

[5] `https://code.google.com/p/ictclas4j/`

[6] `http://www.gnu.org/licenses/`

3 General comments and conclusions

3.1 Comments on the results

LogMap has been one of the top systems in the OAEI 2015 and the only system that participates in all tracks. Furthermore, it has also been one of the few systems implementing repair techniques and providing (almost) coherent mappings in all tracks.

LogMap's main weakness is that the computation of candidate mappings is based on the similarities between the vocabularies of the input ontologies; hence, in the cases where the ontologies are lexically disparate or do not provide enough lexical information LogMap is at a disadvantage.

3.2 Discussions on the way to improve the proposed system

LogMap is now a stable and mature system that has been made available to the community and has been extensively tested. There are, however, many exciting possibilities for future work. For example we aim at improving the current multilingual features and the current use of external resources like BioPortal. Furthremore, we are applying LogMap in practice in the domain of oil and gas industry within the FP7 Optique[7] [18, 15, 10, 17]. This practical application presents a very challenging problem.

Acknowledgements

This work was supported by the EPSRC projects MaSI[3], Score! and DBOnto, and by the EU FP7 project Optique (grant agreement 318338).

We would also like to thank Ian Horrocks, Anton Morant, Yujiao Zhou Weiguo Xia, Xi Chen, Yuan Gong and Shuo Zhang, who have contributed to the LogMap project in the past.

References

1. Agrawal, R., Borgida, A., Jagadish, H.V.: Efficient management of transitive relationships in large data and knowledge bases. In: ACM SIGMOD Conf. on Management of Data. pp. 253–262 (1989)
2. Baeza-Yates, R.A., Ribeiro-Neto, B.A.: Modern Information Retrieval. ACM Press / Addison-Wesley (1999)
3. Chen, X., Xia, W., Jiménez-Ruiz, E., Cross, V.: Extending an ontology alignment system with bioportal: a preliminary analysis. In: Poster at Int'l Sem. Web Conf. (ISWC) (2014)
4. Christophides, V., Plexousakis, D., Scholl, M., Tourtounis, S.: On labeling schemes for the Semantic Web. In: Int'l World Wide Web (WWW) Conf. pp. 544–555 (2003)
5. Cuenca Grau, B., Horrocks, I., Kazakov, Y., Sattler, U.: Modular reuse of ontologies: Theory and practice. J. Artif. Intell. Res. 31, 273–318 (2008)
6. Dowling, W.F., Gallier, J.H.: Linear-time algorithms for testing the satisfiability of propositional Horn formulae. J. Log. Prog. 1(3), 267–284 (1984)

[7] http://www.optique-project.eu/

7. Faria, D., Jiménez-Ruiz, E., Pesquita, C., Santos, E., Couto, F.M.: Towards annotating potential incoherences in bioportal mappings. In: 13th Int'l Sem. Web Conf. (ISWC) (2014)
8. Fridman Noy, N., Shah, N.H., Whetzel, P.L., Dai, B., et al.: BioPortal: ontologies and integrated data resources at the click of a mouse. Nucleic Acids Research 37, 170–173 (2009)
9. Ghazvinian, A., Noy, N.F., Jonquet, C., Shah, N.H., Musen, M.A.: What four million mappings can tell you about two hundred ontologies. In: Int'l Sem. Web Conf. (ISWC) (2009)
10. Giese, M., Soylu, A., Vega-Gorgojo, G., Waaler, A., Haase, P., Jimenez-Ruiz, E., Lanti, D., Rezk, M., Xiao, G., Ozcep, O., Rosati, R.: Optique — Zooming In on Big Data Access. Computer 48(3), 60–67 (2015)
11. Jiménez-Ruiz, E., Cuenca Grau, B.: LogMap: Logic-based and Scalable Ontology Matching. In: Int'l Sem. Web Conf. (ISWC). pp. 273–288 (2011)
12. Jiménez-Ruiz, E., Cuenca Grau, B., Horrocks, I., Berlanga, R.: Logic-based assessment of the compatibility of UMLS ontology sources. J. Biomed. Sem. 2 (2011)
13. Jiménez-Ruiz, E., Cuenca Grau, B., Zhou, Y., Horrocks, I.: Large-scale interactive ontology matching: Algorithms and implementation. In: Europ. Conf. on Artif. Intell. (ECAI) (2012)
14. Jiménez-Ruiz, E., Grau, B.C., Xia, W., Solimando, A., Chen, X., Cross, V.V., Gong, Y., Zhang, S., Chennai-Thiagarajan, A.: Logmap family results for OAEI 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. pp. 126–134 (2014)
15. Jiménez-Ruiz, E., Kharlamov, E., Zheleznyakov, D., Horrocks, I., Pinkel, C., Skjæveland, M.G., Thorstensen, E., Mora, J.: BootOX: Practical Mapping of RDBs to OWL 2. In: International Semantic Web Conference (ISWC) (2015), http://www.cs.ox.ac.uk/isg/tools/BootOX/
16. Jiménez-Ruiz, E., Meilicke, C., Cuenca Grau, B., Horrocks, I.: Evaluating mapping repair systems with large biomedical ontologies. In: 26th Description Logics Workshop (2013)
17. Kharlamov, E., Hovland, D., Jiménez-Ruiz, E., Lanti, D., Lie, H., Pinkel, C., Rezk, M., Skjæveland, M.G., Thorstensen, E., Xiao, G., Zheleznyakov, D., Horrocks, I.: Ontology Based Access to Exploration Data at Statoil. In: International Semantic Web Conference (ISWC). pp. 93–112 (2015)
18. Kharlamov, E., Jiménez-Ruiz, E., Zheleznyakov, D., et al.: Optique: Towards OBDA Systems for Industry. In: Eur. Sem. Web Conf. (ESWC) Satellite Events. pp. 125–140 (2013)
19. Meilicke, C.: Alignment Incoherence in Ontology Matching. Ph.D. thesis, University of Mannheim (2011)
20. Nebot, V., Berlanga, R.: Efficient retrieval of ontology fragments using an interval labeling scheme. Inf. Sci. 179(24), 4151–4173 (2009)
21. Solimando, A., Jiménez-Ruiz, E., Guerrini, G.: Detecting and correcting conservativity principle violations in ontology-to-ontology mappings. In: Int'l Sem. Web Conf. (ISWC) (2014)
22. Solimando, A., Jiménez-Ruiz, E., Guerrini, G.: A multi-strategy approach for detecting and correcting conservativity principle violations in ontology alignments. In: Proc. of the 11th International Workshop on OWL: Experiences and Directions (OWLED). pp. 13–24 (2014)

LYAM++ Results for OAEI 2015

Abdel Nasser Tigrine, Zohra Bellahsene, Konstantin Todorov

{lastname}@lirmm.fr
LIRMM / University of Montpellier, France

Abstract. The paper presents a novel technique for aligning cross-lingual ontologies that does not rely on machine translation, but uses the large multilingual semantic network BabelNet as a source of background knowledge. In addition, our approach applies a novel orchestration of the components of the matching workflow. We demonstrate that our method outperforms considerably the best techniques in the state-of-the-art.

1 Presentation of the system

In spite of the considerable advance that has been made in the field of ontology matching recently, many questions remain open [1]. The current work addresses the challenge of using background knowledge with a focus on aligning cross-lingual ontologies, i.e., defined in different natural languages [2].

Indeed, considering multilingual and cross-lingual information is becoming more and more important, in view particularly of the growing number of content-creating non-English users and the clear demand of cross-language interoperability. In the context of the web of data, it is important to propose procedures for linking vocabularies across natural languages, in order to foster the creation of a veritable global information network.

The use of different natural languages in the concepts and relations labeling process is becoming an important source of ontology heterogeneity. The methods that have been proposed to deal with it most commonly rely on automatic translation of labels to a single target language [3] or apply machine learning techniques [2]. However, machine translation tolerates low precision levels and machine learning methods require large training corpus that is rarely available in an ontology matching scenario. An inherent problem of translation is that there is often a lack of exact one-to-one correspondence between the terms in different natural languages.

1.1 State, purpose, general statement

We present LYAM++ (Yet Another Matcher - Light), a fully automatic cross-lingual ontology matching system that does not rely on machine translation. Instead, we make use of the openly available general-purpose multilingual semantic network BabelNet[1] in order to recreate the missing semantic context in

[1] http://babelnet.org/

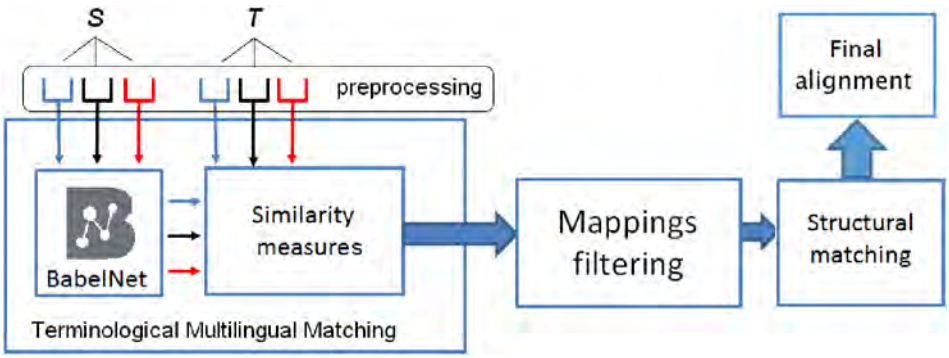


Fig. 1: The processing pipeline of LYAM++.

the matching process. Another original feature of our approach is the choice of orchestration of the matching workflow. Our experiments on the MultiFarm[2] benchmark data show that (1) our method outperforms the best approaches in the current state-of-the-art and (2) the novel workflow orchestration provides better results compared to the classical one.

1.2 Specific techniques used

The workflow of LYAM++ is given in Fig. 1. We take as an input a source ontology S, given in a natural language l_S and a target ontology T, given in a language l_T. The overall processes consists of four main components: a terminological multilingual matcher, a mapping selection module and, finally, a structural matcher. One of the original contributions of this work is the choice of orchestration of these components. Indeed, the places of the mapping selection module and the structural matcher are reversed in the existing OM tools [4]. However, we wanted to ensure that we feed only good quality mappings to the structural matcher, therefore we decided to filter the discovered correspondences right after producing the initial alignment. This decision is supported experimentally in the following section.

The *multilingual terminological matching* module, the second contribution described in this paper, acts on the one hand as a preprocessing component and, on the other hand – as a light-weight terminological matcher between cross-lingual labels. We start by splitting the elements of each ontology in three groups: labels of classes, labels of object properties and labels of data object properties (in colors blue, black and red in the figure), since these groups of elements are to be aligned separately. A standard preprocessing procedure is applied on these sets of labels, comprising character normalization, stop-words filtering, tokenization and lemmatization. The tokens of the elements of T are then aligned to BabelNet. At first, every token of a given label s in S is enriched by related

[2] http://web.informatik.uni-mannheim.de/multifarm/

terms and synonyms from BabelNet and all of these terms are represented in the language l_T, which makes these terms comparable to the tokens of the labels in T. A simple similarity evaluation by the help of the Jaccard coefficient selects the term in each set of related terms corresponding to a given token from s that has the highest score with respect to every token in each label of T. This helps to restitute the label s in the language l_T. Finally, the labels in each group of S and T, seen as sets of tokens, are compared by using the Soft TFIDF similarity measure [5], which produces an intermediate terminological alignment.

The three remaining components are standard OM modules [4], although ordered in a new manner. The *Mapping selection* is a module that transforms the initial 1 to many mapping to a 1:1 alignment based on the principle of iteratively retaining the pairs of concepts with maximal value of similarity. Finally, the *structural matcher* component filters the trustworthy pairs of aligned concepts by looking at the similarity values produced for their parents and their children in the ontology hierarchies.

1.3 Link to the system and parameters file

The system is not yet available online. The reason for that is that it depends heavily on the use of BabelNet, which is a protected source. We are working on implementing a sharable version of LYAM++ making use of different open access background knowledge sources.

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

The alignments produced by LYAM++ for this year's Multifarm track can be found under the following link: `http://www.lirmm.fr/benellefi/Lyam++.rar`

2 Results

We have evaluated our approach on data coming from the ontology alignment evaluation initiative (OAEI)[3] and particularly Multifarm—a benchmark designed for evaluating cross-lingual ontology matching systems. Multifarm data consist of a set of 7 ontologies originally coming from the *Conference* benchmark of OAEI, translated into 8 languages. Two evaluation tasks are defined: *task 1* consists in matching two different ontologies given in different languages, while *task 2* aims to align different language versions of one single ontology.

We have performed experiments on both tasks by using the pairs of languages given in the summary of our results in Table 1.

In another experiment, we have evaluated the results obtained by using our novel orchestration of matching components, as compared to the standard orchestration. The figures in Table 2 show that the workflow proposed in this paper acts in favor of achieving better results as compared to the standard method.

[3] http://oaei.ontologymatching.org/

Table 1: Comparing LYAM++ to AML

Lang. pair	FR-RU	FR-PT	FR-NL	ES-FR	ES-RU	ES-PT	ES-NL	EN-PT	EN-RU	EN-FR
LYAM++	0.54	0.58	0.62	0.60	0.60	0.60	0.63	0.67	0.53	0.59

Average F-measures over all threshold values per language pair for task 1.

Lang. pair	FR-RU	FR-PT	FR-NL	ES-FR	ES-RU	ES-PT	ES-NL	EN-PT	EN-RU	EN-FR
LYAM++	0.58	0.72	0.67	0.77	0.64	0.70	0.68	0.74	0.59	0.85

Average F-measures over all threshold values per language pair for task 2.

Table 2: Comparing the standard and the novel orchestrations

Language pair	EN-FR		EN-RU		ES-FR
Standard (avg)	0.45		0.32		0.39
Novel (avg)	0.84		0.59		0.76

Average F-measures over all threshold values per language pair.

Threshold Value	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Standard (avg)	0.42	0.42	0.42	0.42	0.42	0.39	0.32	0.20
Novel (avg)	0.78	0.78	0.78	0.78	0.78	0.75	0.65	0.4

Average F-measures over all language-pairs per threshold value.

3 Discussions on the way to improve the proposed system

Currently, we are working on enhancing the system in order to make applicable to the general ontology matching problem and not only to cross-lingual ones. We have generated first results on the Conference benchmark without any modification in the settings and our results are quite promising. For the majority of the datasets (ontology pairs) our system achieves a f-score almost as good as the f-score of AML, the best performing system on that track.

We consider that a key feature for the improvement of our system is the appropriate choice of background knowledge. In order to improve the results achieved on the Conference track, we plan to use monolingual general purpose background knowledge (for example, the english subgraphs of YAGO or DBPedia) instead of BabelNet.

We intend to use domain specific background knowledge in order to solve alignment problems in specific areas of knowledge. More precisely, we plan to participate on the Anatomy track by testing different kinds of domain specific background knowledge, such as UMLS or other.

4 Conclusions

We presented an efficient approach for aligning cross-lingual ontologies by using the multilingual lexical database BabelNet. Subjects of ongoing and future work are (1) testing and evaluating different sources of external knowledge, (2) applying the approach to a larger set of languages and (3) adaptation of the

approach to the monolingual case and studying the use of background knowledge in a monolingual ontology matching scenario.

References

1. P. Shvaiko and J. Euzenat, "Ontology matching: state of the art and future challenges," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 25, no. 1, pp. 158–176, 2013.
2. D. Spohr, L. Hollink, and P. Cimiano, "A machine learning approach to multilingual and cross-lingual ontology matching," in *The Semantic Web–ISWC 2011*, pp. 665–680, Springer, 2011.
3. D. Faria, C. Pesquita, E. Santos, M. Palmonari, I. F. Cruz, and F. M. Couto, "The agreementmakerlight ontology matching system," in *On the Move to Meaningful Internet Systems: OTM 2013 Conferences*, pp. 527–541, Springer, 2013.
4. D. Ngo, Z. Bellahsene, and K. Todorov, "Opening the black box of ontology matching," in *The Semantic Web: Semantics and Big Data*, pp. 16–30, Springer, 2013.
5. W. W. Cohen, P. D. Ravikumar, and S. E. Fienberg, "A comparison of string distance metrics for name-matching tasks," in *IIWeb*, pp. 73–78, 2003.
6. D. Faria, C. Martins, A. Nanavaty, A. Taheri, C. Pesquita, E. Santos, I. F. Cruz, and F. M. Couto, "Agreementmakerlight results for OAEI 2014," in *Procs of the 9th Intl Workshop on Ontology Matching (ISWC)*, pp. 105–112, 2014.

MAMBA - Results for the OAEI 2015

Christian Meilicke

Research Group Data and Web Science
University of Mannheim, 68163 Mannheim, Germany
christian@informatik.uni-mannheim.de

1 Presentation of the system

Most matching systems implement their functionality as a sequential process. Such systems start with analyzing different types of evidence, in most cases with a focus on the involved labels, and generate, as an intermediate result, a set of weighted matching hypotheses. From the intermediate result a subset of the generated hypotheses is chosen as final output. The approach implemented in MAMBA differs significantly from this approach.

MAMBA[1] treats labels (and their parts) as well as logical entities (classes and properties) as first class citizens in an optimization problem. During the matching process MAMBA generates hypotheses about equivalences between labels and tokens, while at the same time mappings between concepts and properties are considered to be true and wrong. MAMBA uses Markov Logic [6] to define constraints that ensure that the underlying assumptions about equivalent tokens are always consistent and that dependencies between labels and entities described by these labels are taken into account. The approach implemented in MAMBA has been described in details in a paper [4] that can also be found in the proceedings of the Ontology Matching Workshop. To avoid redundancy, we omit a description of the underlying approach in this paper. Instead of that we comment on some results and discuss open issues.

MAMBA is available at http://web.informatik.uni-mannheim.de/mamba/. Note that MAMBA was developed with the motivation to illustrate the benefit of the approach roughly sketched in [3] and finally presented in [4]. Thus, MAMBA is not a general-purpose ontology matching system but a research prototype.

2 Results

2.1 Conference Track

The OAEI conference track was used as one of the main test sets used during the development and testing of MAMBA. The achieved results are shown in Table 1.

Comparing these results against the results of previous OAEI editions, MAMBA is always among the best two systems with respect to F-measure. Only the system YAM++ [1] achieved an F-measure of .71 (ra-2) and .74 (ra-1), which is a bit better than the results of MAMBA.

[1] MAMBA stands for **Ma**nnheim **M**atcher based on a **B**ilayered **A**pproach

Gold Standard	Precision	F-measure	Recall
ra-1	.80	.68	.59
ra-2	.83	.72	.63

Table 1. Results for the Conference track

2.2 Results for the other tracks

Due to the fact that MAMBA is currently only a research prototype mainly developed for testing the approach that we described in [4], we have not conducted many experiments on other data sets. However, we already know that MAMBA will probably not be able to match ontologies with more than 1000 concepts due to the underlying optimization problem. Furthermore, we made only a very quick test with the bibliographic benchmark, to ensure that the basic functionality of a matching system is implemented.

3 General comments

3.1 Comments on the Results

The results for the Conference track illustrate the benefits of the proposed approach. Note that we applied a very restrictive approach for computing the input similarities which are used as evidence for the equivalence hypotheses between the tokens. We used more or less the maximum of Levensthein similarity and Wu Palmer WordNet similarity together with a very simple method for generating similarities between pairs of tokens that contain abbreviations (e.g., *ProgramCommitteeMember* vs. *PCMember*). Most approaches use a richer set of method with a fine tuned aggregation method. Thus, we believe that the results of MAMBA can be improved by using better similarity measures.

We did not compute results for any other track. While we were mainly interested in understanding the impact of our new approach, we could spend only a limited time in checking whether MAMBA is capable of generating alignments for all kind of input ontologies that might differ in format and in the way how labels are used to describe the logical entities. Preliminary experiments with one test set from the benchmark series showed that MAMBA generates for these synthetic data sets only mediocre results.

The most critical issues are related to the runtimes of MAMBA. MAMBA will not terminate for ontologies with more than 1000 concepts. The optimization problem that needs to be solved is NP-hard. Note also the the runtime performance of MAMBA is even worse than the runtime performance of CODI [2], which also defines internally an optimization problem. Due to the two layers of tokens and entities, MAMBA translates a matching problem into a more complex problem with more variables and more constraints.

3.2 Improving the Approach

An additional amount of engineering work is required to make MAMBA more robust. There is a high chance that the current version contains several bugs that need to be

detected via extensive testing. We know, for example, that complex domain and range restrictions are currently not correctly interpreted by MAMBA.

The runtime problems of MAMBA cannot be solved easily. We are currently using a stack of systems (Rockit [5], GUROBI), where each system is known to be one of the most efficient systems for solving the type of problems that MAMBA generates. Moreover, we apply already a specific technique to speed up the matching task, by first solving a relaxed version of the matching problem, which allows to solve the harder problem more efficiently.[2]

Our main motivation while developing MAMBA was to show the need for generating alignments that are consistent with respect to the corresponding assumptions about the meanings of the involved tokens. This general idea is not necessarily bound to the use of optimization techniques. Greedy techniques can also be used to ensure this special kind of label/entity alignment consistency. Indeed, such approaches have to be used to make the general idea applicable to matching larger ontologies as we find them in the Anatomy track or in the Large Biomedical track.

3.3 Comments on OAEI test cases

The availability of the OAEI test cases has revealed that MAMBA needs to be significantly improved to become a robust matching systems instead of being just a set of scripts that have been used to illustrate the benefits of a specific approach. We must admit that we underestimated the engineering work that is required to implement these improvements.

However, our sole focus on the conference track was mainly motivated by the fact that the conference track is the only track that has a manually generated, high quality gold standard that is at the same time easily understandable, while the ontologies are relatively expressive and differ partially in their modeling style. This real world scenario results in a great deal of non trivial mappings that our approach is designed to detect. For that reasons it would be a significant improvement if the OAEI would offer a second track that has a similar characteristic as the conference track.

4 Conclusion

MAMBA is our attempt to implement the approach described in [4] as a matching system. While we were able to generate good results for the test cases of the Conference track, we have not yet systematically tested the performance of MAMBA for the other tracks. We already know that MAMBA will not terminate in acceptable time for test cases with more than 1000 classes. Nevertheless, the good results that we achieved for the conference track might be a motivation to modify existing matching systems in a way that the resulting mappings are consistent with respect to the implicit assumptions regarding the equivalence of the involved tokens.

[2] Unfortunately, this approach is not even explained in [4]. Contact the author if you are interested in the details.

References

1. Ngo Duyhoa and Zohra Bellahsene. Yam++ results for oaei 2013. In *Proceedings of the 8th International Workshop on Ontology Matching (OM 2013)*, 2013.
2. Jakob Huber, Timo Sztyler, Jan Noessner, and Christian Meilicke. Codi: Combinatorial optimization for data integration–results for oaei 2011. *Ontology Matching*, 134, 2011.
3. Christian Meilicke, Jan Noessner, and Heiner Stuckenschmidt. Towards joint inference for complex ontology matching. In *AAAI (Late-Breaking Developments)*, 2013.
4. Christian Meilicke and Heiner Stuckenschmidt. A new paradigm for alignment extraction. In *Proceedings of the Tenth International Workshop on Ontology Matching (OM 2015)*, 2015.
5. Jan Noessner, Mathias Niepert, and Heiner Stuckenschmidt. RockIt: Exploiting parallelism and symmetry for map inference in statistical relational models. 2013.
6. Matthew Richardson and Pedro Domingos. Markov logic networks. *Machine learning*, 62(1-2):107–136, 2006.

RiMOM Results for OAEI 2015

Yan Zhang, Juanzi Li

Tsinghua University, Beijing, China.
z-y14@mails.tsinghua.edu.cn ljz@ keg.tsinghua.edu.cn

Abstract. This paper presents the results of RiMOM in the Ontology Alignment Evaluation Initiative (OAEI) 2015. We only participated in Instance Matching@OAEI2015. We first describe the overall framework of our matching System (RiMOM); then we detail the techniques used in the framework for instance matching. Last, we give a thorough analysis on our results and discuss some future work on RiMOM.

1 Presentation of the system

As the infrastructure of the Semantic Web, knowledge base has become a dominant mechanism to represent the data semantics on the Web. In this circumstance, a large number of ontological knowledge bases have been built and published, such as DBpedia[1]. , YAGO [2], Xlore [3], etc. In real environment of the Semantic Web, data is always distributed on heterogeneous data sources (ontology). It is inevitable that the knowledge about the same real-world entity may be stored in different knowledge bases. Therefore, there is a growing need to align different knowledge bases so that we can easily get complete information that we are interested in.

Some good results have been achieved in the field of ontology matching [4]. Previous researches always focus on aligning the schema elements (i.e. concepts and properties) in knowledge bases. Most recently, with the rapid development of semantic web, there have been many large-scale ontologies which contain millions of entities. It is obviously that the number of instances is much larger than other elements (e.g. concepts and properties) in these ontologies. For example, the DBpedia contains 882,000 instances of 6 main concepts. Thus, the large-scale instance matching has become the key point in the ontology matching system.

Different from the schema matching, the instance matching always has the following characteristics:

1. The number of instances may be enormous.
2. The schema is straightforward.
3. In practice, the knowledge base is always updated dynamically.

In consideration of these differences, we proposed a large-scale instance matching system, RiMOM.

There are two major techniques in our system, inverted index and multi-strategy:

1. We index the instances based on their objects in two knowledge bases respectively, and then select the instances which contain the same keys as candidate instance

pairs. We limit the number of pairs to be compared by this step, which significantly improve the efficiency of the system.
2. We implement several matchers in our instance matching system, we can execute these matchers in parallel and then aggregate the result according to the characteristics of the source ontologies.

In order to solve the challenges in large-scale instance matching, we propose an instance matching framework RiMOM-2015 (RiMOM-Instance Matching), which is based on our former ontology matching system RiMOM [5]. The RiMOM-2015 framework is designed for large-scale instance matching task specially. It presents a novel multi-strategy method to be fit for different kind of ontology and employs inverted index to imporve the efficiency.

1.1 State, purpose, general statement

This section describes the overall framework of RiMOM. The overview of the instance matching system is shown in Fig. 1. The system includes seven modules, i.e., *Preprocess*, *Predicate Alignment*, *Mathcher Choosing*, *Candidate Pair Generation*, *Matching Score Calculation*, *Instance Alignment* and *Validation*. The sequences of the process are shown in the Fig. 1. We illustrate the process as follows.

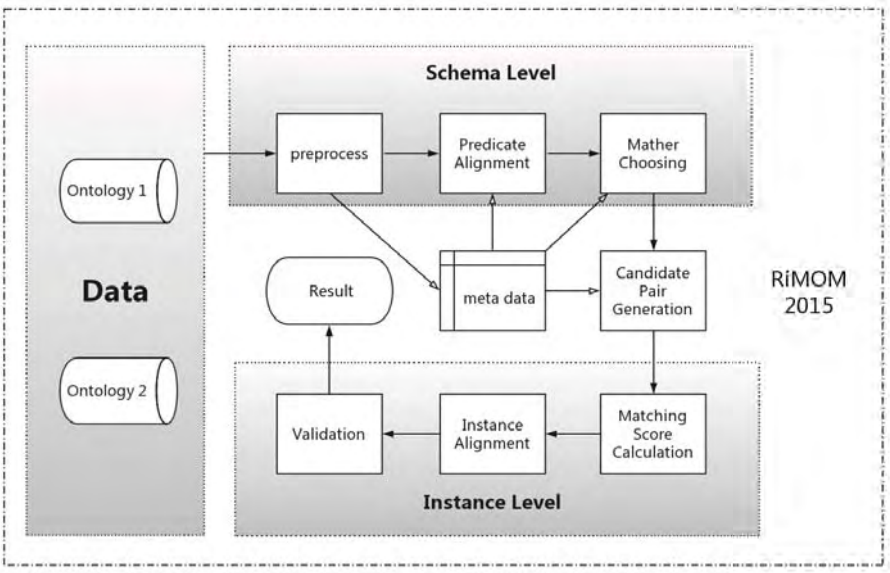


Fig. 1. Framework of RiMOM 2015

1. **Preprocess:** The system begins with *Preprocess*, which loads the ontologies and parameters into system. In the meantime, preprocessor can get some meta data about the two ontologies, which will be used in the later processes, *Predicate alignment* and *Matcher choosing*
2. **Predicate Alignment:** In this process, we will get the alignments of the predicates between the two ontologies. Currently, in our system, this process is semi-automatic.
3. **Matcher choosing:** The system will choose the most suitable one or more matchers according to the meta data of the ontologies.
4. **Candidate Pairs Generation:** In this step, we get the candidate pair when the instances have the same literal objects on some discriminatory predicate.
5. **Matching Score Calculation:** After the candidate set generation, we calculate more accurate similarity using the algorithm chosen by step 3. In this task, the vector distance similarity was calculated between each candidate pair.
6. **Instance Alignment:** According to the similarity calculated in step 5, we get the final instance alignment.
7. **Validation:** We will evaluate the alignment result on Precision, Recall and F1-Measure if there is validation data set.

1.2 Specific techniques used

This year we only participate in the Instance Matching track. We will describe specific techniques used in this track.

Data Preprocessing: First, we remove some stop words like "a, of, the", etc. Afterwards, we calculate the TF-IDF values of words in each knowledge base. We also calculate some information of each predicate, in order to find the important predicates.

Predicate Alignment: It is apparent that we should get the alignment of the predicates before we calculate the similarity of instances. The predicates can express rich semantics, and there exists one-to-one, one-to-many, or many-to-many relationships among these predicates. We can find some of one-to-one relationships through calculating the *Jaccard Similarity* of the two predicates. i.e.

$$sim(p_i, p_j) = \frac{|O_{p_i} \cap O_{p_j}|}{|O_{p_i} \cup O_{p_j}|}$$

where p_i and p_j are predicates in two ontologies respectively. O_{p_j} is the $range$ of the predicate p_j.

There are also some one-to-many relationships. We get the alignments of them by manual regulations, e.g.

$$object(p_i) = \sum_{j=1}^{n} object(p_j)$$

$$object(p_i) = \max_{j=1..n} object(p_j)$$

$$object(p_i) = \min_{j=1..n} object(p_j)$$

Candidate Pairs Generation: This step aims to pick a relatively small set of candidate pairs from all pairs. Due to the large scale of knowledge bases, it is impossible to calculate matching scores of all instance pairs. In our method, we firstly generate the inverted index on the objects. instance pairs are selected into the candidate set when they have common objects. This method may reduce the recall, but it also reduce the scale of computation significantly.

Multi-Strategy: We implement several matchers in our system, e.g. label-based approach and structure-based approach. In the preprocess step, we will compare the schema of the two ontologies. If the range of predicates is similar, the label-based approach will play a key role in the matching process. Otherwise, the literal properties are not similar (e.g. the two ontologies are defined in different languages), label-based approach will not be effective. In this case, we will get some supplementary information (e.g. machine translation, WordNet), or use structure-based appraoch.

Similarity Calculation: In OAEI 2015 instance matching track, the ontologies are all defined in the same language, English. In the tasks which we took part in, $author - dis$ and $author - rec$, the schema of the ontologies tend to be similar. So label-based vector distance matcher is chosen to calculate the similarity of the instances, it is defined as follows:

$$L_a = Objects(I_a)$$

where I_a is an instance, L_a is a list which contains all of the objects of the instance I_a.

$$Sim(I_a, I_b) = Sim(L_a, L_b) = \frac{1}{|L_a|} \sum_{O_a \in L_a} \max(Sim(O_a, O_b)|O_b \in L_b)$$

where O_a is one of the objects in the list L_a. We define the similarity of the two instances equals to the similarity of their objects list. For each O_a in L_a, we find a most similar object O_b in L_b. The algorithm varies with the data type of the object. For example, for date, we use the indicator function. The indicator function will be 1 when the dates are the same, otherwise, 0. For some literal properties, such as "title", we compute cosine similarity based on the tf-idf vectors.

Instance Alignment After we get the accurate similarity, for each instance in source ontology, we choose the instance which has the best score in target ontology. Then we filter the result on a certain threshold and get the final Instance Alignment.

1.3 Link to the system and parameters file

The RiMOM system (2015 version) can be found at `https://www.dropbox.com/s/6bx4pb46ytvddvy/RiMOM.zip?oref=e`.

2 Results

The Instance Matching track contains five subtasks. we present the results and related analysis for the two subtasks (author-disambiguation and author-recognition) in the following subsections.

2.1 Author Disambiguation sub-task

The goal of the author-dis task is to link OWL instances referring to the same person (i.e., author) based on their publications. We can use the Sandbox (small scale data set) to tune our parameters. The class 'author' have only one literal properties, 'name'. So we must get alignments on the class 'publication'. Finally, we get 854 pairs for Sandbox task, and 8428 pairs for Mainbox task.

	Expected mappings	Retrieved mappings	Precision	Recall	F-measure
EXONA	854	854	0.941	0.941	0.941
InsMT+	854	722	0.834	0.705	0.764
Lily	854	854	0.981	0.981	0.981
LogMap	854	779	0.994	0.906	0.948
RiMOM	854	854	0.929	0.929	0.929

Table 1. The result for Author-dis sandbox

	Expected mappings	Retrieved mappings	Precision	Recall	F-measure
EXONA	8428	144827	0	0	NaN
InsMT+	8428	7372	0.76	0.665	0.709
Lily	8428	8428	0.964	0.964	0.964
LogMap	8428	7030	0.996	0.831	0.906
RiMOM	8428	8428	0.911	0.911	0.911

Table 2. The result for Author-dis mainbox

The reference alignments of sandbox are provided by sponsor, so we only pay attention to mainbox. As shown in table 2, the results for the author-dis mainbox task are: Precision 0.911, Recall 0.911, F-measure 0.911, which is slightly lower than sandbox. Afterwards, we find that the property 'title' plays a key role in publication. So we think that we can get a better result if we do some deeper work on it.

2.2 Author Recognition sub-task

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. The final goal is similar with the Author-dis task, but there are some changes on schema of the ontology. The most remarkable is that there exists one-to-many relationships between the properties. So we add some manual regulation to solve the problem.

As show in table 4, RiMOM get a excellent result on author-rec task. The results for the author-dis mainbox task are: Precision 0.999, Recall 0.999, Fmeasure 0.999, which expresses that the algorithm we implement is very suitable for this task.

	Expected mappings	Retrieved mappings	Precision	Recall	F-measure
EXONA	854	854	0.518	0.518	0.518
InsMT+	854	90	0.556	0.059	0.106
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

Table 3. The result for Author-rec sandbox

	Expected mappings	Retrieved mappings	Precision	Recall	F-measure
EXONA	8428	8428	0.409	0.409	0.409
InsMT+	8428	961	0.246	0.028	0.05
Lily	8428	8424	0.999	0.998	0.999
LogMap	8428	8436	0.999	1.0	0.999
RiMOM	8428	8428	0.999	0.999	0.999

Table 4. The result for Author-rec mainbox

2.3 Discussions on the way to improve the proposed system

Our system need align the predicates before instance matching, and in this process, the system is required to scan all of the instances in the ontology, which may cause a waste of time. In addition, the process of $PredicateAlignment$ is semi-automatic, we have to add some manual regulations to deal with the one-to-many relationships.

In conclusion, we hope to develop our system through inventing an algorithm to align the predicates automatically and iteratively. Firstly we can use the values of predicates to align the instances, and in turn, the aligned instances will help us to update the similarity for predicates. In this way, we will gradually get the final alignment result.

2.4 Comments on the OAEI 2015 measures

These two tasks are instance matching task on publication data set. We use the reference of the sandbox to tune the parameters,and it turns out that our approach is effective. We also find that the inverted index not only improve efficiency, but reduce the mistake and increase the Precision. There are also some aspects we are not satisfied with. For time's sake, we don't take part in other three tasks. Finally, we are looking forward to making some progress in the next OAEI campaign.

3 Conclusion and future work

In this paper, we present the system of RiMOM in OAEI 2015 Campaign. We participate in intance matching track this year. We described specific techniques we used in the task. In our project, we design a new framework to deal with the instance matching task. The result turns out that our method is effective and efficient.

In the future, we will develop an iterative algorithm to align the predicates automatically.

References

1. Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., Hellmann, S.: Dbpedia - A crystallization point for the web of data. J. Web Sem. **7**(3) (2009) 154–165
2. Hoffart, J., Suchanek, F.M., Berberich, K., Weikum, G.: YAGO2: A spatially and temporally enhanced knowledge base from wikipedia. Artif. Intell. **194** (2013) 28–61
3. Wang, Z., Li, J., Wang, Z., Li, S., Li, M., Zhang, D., Shi, Y., Liu, Y., Zhang, P., Tang, J.: Xlore: A large-scale english-chinese bilingual knowledge graph. In: Proceedings of the ISWC 2013 Posters & Demonstrations Track, Sydney, Australia, October 23, 2013. (2013) 121–124
4. Shvaiko, P., Euzenat, J.: Ontology matching: State of the art and future challenges. IEEE Trans. Knowl. Data Eng. **25**(1) (2013) 158–176
5. Li, J., Tang, J., Li, Y., Luo, Q.: Rimom: A dynamic multistrategy ontology alignment framework. IEEE Trans. Knowl. Data Eng. **21**(8) (2009) 1218–1232

RSDL Workbench Results for OAEI 2015*

Simon Schwichtenberg and Gregor Engels

University of Paderborn, s-lab – Software Quality Lab, Germany
{simon.schwichtenberg,engels}@upb.de

Abstract The vision of automatic service composition is to automatically combine single services to a software solution that satisfies certain requirements. Comprehensive service specifications are needed to receive suitable compositions. The Rich Service Description Language (RSDL) has been developed and can be used to specify ontological and behavioral semantics of services comprehensively. Part of a service's RSDL specification is its domain ontology that comprises concepts to describe, e.g., the service's input and output parameters. The RSDL Workbench (RSDLWB) is a platform that provides tools for the specification, matching, and composition of services. In particular, RSDLWB matches ontologies that are part of RSDL specifications. In this paper, we present that ontology matcher and the evaluation results as determined by the Ontology Alignment Evaluation Initiative (OAEI). Compared to the last campaign, we improved the runtime while maintaining the quality level of the produced alignments.

1 Presentation of the system

RSDLWB is a collection of tools for the specification, matching, and automatic composition of services. On the one hand, service requesters need to specify *service requests*, i.e., the requirements for services they need. On the other hand, service providers need to specify their *service offers*, i.e., the services they provide. Comprehensive, multi-faceted specifications that describe structural as far as behavioral aspects are needed to determine proper service compositions. A RSDL specification of a service defines its individual ontology and operation signatures. Besides these structural aspects of a service, the specifications also comprises behavioral aspects as pre- and postconditions of operations and operation protocols.

The ontologies describe the concepts and relations that appear in the domain of a service, e.g. to describe parameter types of operations. Within this paper, only ontologies that are part of service specifications are in the focus. Comprehensive specifications can be created in languages like RSDL [4], which is similar to the Web Ontology Language for Services (OWL-S).

The task of matching requests and services is called *Service Discovery*. For the matching of multi-faceted specifications, multiple matchers are needed, while each is specialized for either the matching of ontologies, operations, or protocols [4].

Since service specifications are created independently, the ontologies they contain are most likely to be heterogeneous in terms of their terminology or conceptualization.

* This work was partially supported by the German Research Foundation (DFG) within the Collaborative Research Centre "On-The-Fly Computing" (SFB 901)

Two ontologies might contain equivalent concepts, while both use different labels or logical hierarchies. The task of ontology matching is to find correspondences between concepts in the ontologies of service requests and offers. Ontology matchers produce *ontology alignments*, i.e., sets of mappings.

RSDLWB also enables the transformation of individuals from one ontology to another, based on the previously calculated ontology alignment. In this context, individuals are instances of the classes defined in an ontology. Within the RSDL specification of a service, the pre- and postconditions of its operations are denoted by Visual Contracts (VCs) [2], i.e., a variant of graph grammar rules. Each rule consists of a Left-Hand Side (LHS) and a Right-Hand Side (RHS). The LHS and RHS of the graph grammar rules are instance graphs that conform to the service's individual ontology. The LHS is the precondition that must hold before the operation can be executed, whereas the RHS describes the effects of the execution. In the short notation of VCs, instances that only appear on the LHS are deleted and marked in red, instances that only appear on the RHS are created and marked in green, and instances that appear on both sides are preserved and marked in black.

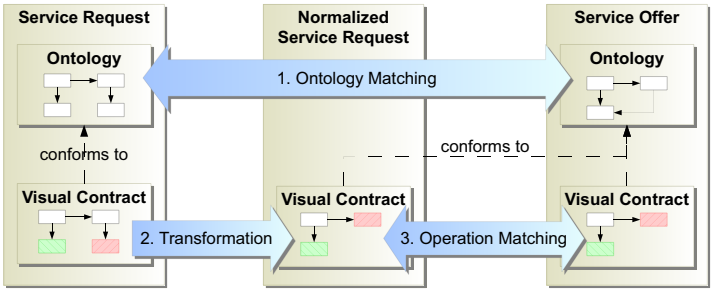


Figure 1: Matching Process [8]

Ontology matching is a prerequisite for the operation matcher that is described in [4]. This matcher requires that specifications conform to the same ontology. Consequently, the heterogeneous ontologies that are contained in request and offer specifications have to be normalized so that operation matching can be applied. A unique feature of RSDLWB is that it produces ontology alignments in terms of relational Query View Transformation (QVT) model transformation scripts. These transformation scripts can be used as a basis to normalize specifications, i.e., to reconcile VCs so that they conform to the same ontology. The relationship between ontology and operation matching is shown in Fig. 1: In a first step, two ontologies are matched and a transformation script is produced. The input of the transformation are the VCs of the request and the output are the corresponding VCs that conform to the ontology of the offer. These normalized VCs are used for the operation matching.

1.1 State, purpose, general statement

The purpose of RSDLWB's ontology matcher is to match ontologies that are part of (RSDL) service specifications. RSDLWB is still under development and continuous improvement. In its current shape, the matcher supports the following OAEI tracks:

benchmark, *anatomy*, *conference*, and *largebio*. The focus for this year's OAEI campaign was to improve the runtime performance of the matcher.

1.2 Specific techniques used

For the OAEI 2015 campaign, a new version of RSDLWB's ontology matcher was introduced that is specialized for OAEI. This version includes the following major changes: (1) Unnecessary time for the conversion of different model representations was eliminated. In particular, the abstraction layer that translates Web Ontology Language (OWL) ontologies to their Ecore representation was removed, so that OWL can be processed directly without an adapter. Furthermore, the matcher does not implement the EMFCompare API[1] anymore, but implements the Semantic Evaluation At Large Scale (SEALS) API[2] directly. (2) In order to avoid quadratic runtime complexity, the matching algorithm does not create a complete similarity matrix to check all possible concept pairs anymore. Instead, a simple heuristic was used as explained in Sect. 1.2. (3) Machine learning techniques were applied to obtain a classifier that can match concept pairs. This classifier was trained on the basis of reference alignments provided by the OAEI tracks.

Algorithm The RSDL Workbench matches `Classes`, `DataProperties`, and `ObjectProperties` independently. At first, a pre-processing normalizes the labels of the concepts. The labels are split into tokens at uppercase characters (CamelCase) or special delimiters like underscores. Each single token is *normalized* by lowercasing and suppression of non-alphabetical characters. These single tokens are concatenated with an underscore to form the normalized label. For example, the concept `OrganizingCommittee` becomes `organizing_committee` or `Positive_Review` becomes `positive_review`.

In a next step, *seed pairs* are selected, i.e., concept pairs that are likely to match. Seed pairs are selected by two different heuristics. The first heuristic selects seed pairs that have identical normalized labels. When identical normalized labels are not available by a sufficient amount, a different heuristic is used. This second heuristic selects seed pairs of concepts that are on the same hierarchy level in respect to the ontology they are defined in. Roots are on level 0, the roots' subclasses are on level 1, and so forth.

Next, the seed pairs are classified by means of the classifier that is described in the following sections. A post-processing ranks the positively classified pairs descending by an aggregated similarity value, i.e., the Euclidean distance of the feature vectors that are described in the following.

Starting from the most similar pairs, these pairs are added to the output alignment in a greedy manner. When a pair $\langle c_1, c_2 \rangle$ is added to the alignment, all other pairs that contain either c_1 or c_2 are discarded. Consequently, the matcher produces only 1:1 mappings.

Machine Learning Features Machine learning classification relies on statistical patterns found in features of example objects. RSDLWB's classifier classifies whether pairs of

[1] `https://www.eclipse.org/emf/compare/`
[2] `http://www.seals-project.eu/`

concepts do match or not. The classification is conducted on the basis of feature vectors. In particular, these feature vectors comprise several similarity values. We experimented with different features that were developed with runtime performance in mind. The features that had been used to train the classifier are described in this paragraph. Further features based on background knowledge are planned for future work.

Hierarchy level similarity relates the hierarchy position of two concepts with respect to the ontology they are defined in. The intuition is that concepts that are located on the similar hierarchy level are similar.

Outdegree similarity relates the outdegree of two concepts. An ontology is a directed triple graph, where each triple (s, p, o) consists of a subject s, predicate p, and an object o. The outdegree of concept c is the number of triples where c is the subject and p and o are free variables. The intuition is that concepts that have a similar number of outgoing edges are similar.

Property count similarity relates the number of properties of two concepts. The intuition is that concepts that have a similar number of properties are similar.

Shared token similarity relates the set of tokens from the labels of two concepts. The Jaccard coefficient is calculated for these token sets. The intuition is that concepts whose labels share many tokens are similar.

Property shared token similarity relates the set of tokens of the labels of all direct property labels of two concepts: At first, all direct properties of a concept are determined. Their labels are split into token sets. Then the Jaccard coefficient is calculated for these token sets. The intuition is that the property labels of two similar concepts share many tokens.

Neighborhood shared token similarity relates the set of tokens of the labels of all direct neighbors of two concepts. A neighbor n of concept c is determined by the triple (c, p, n), where p is a free variable. The Jaccard coefficient is calculated for these token sets. The intuition is that the token sets created from the labels of all adjacent neighbors of two similar concepts have a high overlap.

Token count similarity relates the number of tokens of the labels of two concepts. The intuition is the labels of similar concepts have a similar amount of tokens.

Substring length similarity relates the string length of two concept labels. If the label of a concept c_1 is contained in label of another concept c_2, this similarity is defined as the quotient of c_1's and c_2's label length. Otherwise, the similarity is 0. The intuition is that the longer the common character sequence is, the more similar the concepts are.

Equivalent shared token similarity relates the set of tokens of the labels of all equivalents of two concepts: At first, all equivalents of a concept are determined according to the *#equivalentClass* relation. Their labels are split into token sets for which the Jaccard coefficient is calculated afterwards. The intuition is that the token sets created from all equivalent classes of similar concepts have a high overlap.

Corpus and Classifier Creation In order to train classifiers with machine learning techniques, positive and negative examples were needed, in which statistical patterns are found that allow distinguishing correct from incorrect matches. A corpus is a set of positive and negative examples and is divided into a training and a validation set. An

example is a vector of feature values for the concept pair $\langle c_1, c_2 \rangle$ plus a matching class, which determines if the concept pair is a correct mapping or not.

Two corpora had been created for each of the OAEI tracks *benchmark*, *anatomy*, *conference*, and *largebio*: One corpus for class and another for property matching. The set of positive examples $I_\oplus$ is the set of mappings that are included in the given reference alignment R_{O_1,O_2}:

$$I_\oplus := R_{O_1,O_2}$$

In contrast to $I_\oplus$, the set of negative examples $I_\ominus$ had to be generated. Randomly generated incorrect pairs are likely to differ a lot from correct pairs, i.e., the values of their feature vectors deviate a lot. Consequently, correct and incorrect pairs can be easily distinguished. However, it is more meaningful to train a classifier on examples that show the subtle differences between correct and incorrect pairs. That is the reason why the set of incorrect pairs $I_\ominus$ was generated depending on $I_\oplus$: Originating from a correct pair $\langle c_1, c_2 \rangle \in I_\oplus$, incorrect pairs were selected from the Cartesian product of c_1's and c_2's direct subclasses. The idea is that c_1's and c_2's subclasses are similar, because c_1 and c_2 form a correct pair. Let $\langle c_1, c_2 \rangle \in I_\oplus$. S_{c_i} is the set of direct subclasses of c_i and $S'_c := S_{c_i} \cup \{c_i\}$. Incorrect pairs are selected from the Cartesian product $S'_{c_1} \times S'_{c_2}$.

$$I_\ominus := S'_{c_1} \times S'_{c_2} \backslash I_\oplus \quad \text{for all } \langle c_1, c_2 \rangle \in I_\oplus$$

The number of generated negative examples was limited by the number of the given positive examples in order to receive balanced sets of positive and negative examples. All examples were distributed by a 66/33 percentage ratio over the training and validation set.

Tab. 1 shows a short evaluation of the quality of the previously described features. In particular, the information gain metric [6] was calculated on the basis of *anatomy*, *benchmark*, *conference*, and *largebio* corpora for class matching. In addition, the average score across all tracks is given.

Feature	Information gain				
	anatomy	benchmark	conference	largebio	∅
#Examples	6958	3032	346	51844	
Substring length similarity	.1224	.3276	.3768	.1484	.2438
Shared token similarity	.1227	.1670	.2812	.1426	.1784
Equivalent shared token similarity	.1227	.1670	.2812	.1426	.1784
Neighborhood shared token similarity	.0957	.1504	.0713	.0810	.0996
Outdegree similarity	.0235	.0830	0	.0229	.0852
Token count similarity	.0749	.0234	.0857	.0251	.0523
Hierarchy level similarity	0	.0968	0	.0476	.0361
Property count similarity	0	.1395	0	0	.0349
Property shared token similarity	0	.0437	.0338	0	.0194

Table 1: Information Gain of Features for Class Matching

In its current shape, RSDLWB uses a Random Forest classifier [1] that was trained on the benchmark corpora. An inclusion of other classifiers trained on the other corpora is planned for future work. The tool suite WEKA [3] was used to create the classifiers.

2 Results

This section first describes the experimental set-up of the different OAEI tracks, in which RSDLWB has been evaluated. The evaluation results regarding RSDLWB are summarized in Tab. 2. The values for precision, F-measure, and recall were calculated with respect to the reference alignments specified in the second column. A detailed explanation of the reference alignments can be found in the respective paragraphs. The harmonic mean of all test cases is stated for *conference* and *multifarm*. Regarding *anatomy* and *largebio*, results for the single test cases are provided particularly.

benchmark The test cases of the *benchmark* track are systematically generated from two seed ontologies – biblio and IFC4 – by modifying or discarding several ontology features. Due to unverified technical difficulties during the execution performed by the organizers, RSDLWB did not produce any alignments for the *benchmark* track.

anatomy The task of the *anatomy* track is to match the Adult Mouse Anatomy and a part of the National Cancer Institute Thesaurus (NCI) describing the human anatomy. With regard to precision, F-measure, and recall, RSDLWB performs similar to the baseline algorithm StringEquiv. RSDLWB achieved high precision but low recall. Compared to the last year's evaluation [7], the quality of the produced alignments stayed approximatively the same. The runtime was improved from 1337 to 22 seconds.

conference This track consists of 16 heterogeneous ontologies in the domain of conference organization. There are three kinds of reference alignments for each test case: ra1, ra2, and rar2. The reference alignment ra2 is the transitive closure of ra1, in which conflicting correspondences had been eliminated by the organizers. The reference alignment rar2 is a refinement of ra2 in which violations had been removed by logical reasoning. Three evaluation modalities are provided for each reference alignments: M1 contains only classes, M2 only properties, and M3 is the union of M1 and M2.

RSDLWB showed its best accuracy regarding the M1 reference alignments. Regarding F-measure and ra1-M1, RSDLWB is better than the baseline algorithm StringEquiv. For ra2-M1 and rar2, RSDLWB is even better for the baseline algorithm edna regarding F-measure. The results for the M2 reference alignments show that RSDLWB matching of properties is improvable. This has also a negative effect on the results for the M3 reference alignments: Compared to the the OAEI 2014 campaign, RSDLWB's accuracy was significantly reduced in respect to ra2-M3. In particular, precision decreased by 0.53, recall increased by 0.02, and F-measure decreased by 0.24. The modalities M1 and M2 cannot be compared, because they were not available for the OAEI 2014 campaign.

multifarm The goal of the *multifarm* track is to evaluate the ability of a matcher to deal with ontologies in different languages. This track has two kinds of tasks: The first kind matches the same ontology in different languages (same) and the second matches different ontologies in different languages (diff).

RSDLWB does not support other languages than English yet. For *multifarm* RSDLWB uses the hierarchy level heuristic as described in Sect. 1.2. This heuristic works

Track	Reference Alignment	Runtime [h:m:s]	Precision	F-measure	Recall
anatomy	Mouse-NCI	00:00:22	.959	.732	.592
conference	H-Mean (ra1-M1)	n/a	.88	.66	.53
conference	H-Mean (ra1-M2)	n/a	.03	.05	.24
conference	H-Mean (ra1-M3)	n/a	.25	.33	.49
conference	H-Mean (ra2-M1)	n/a	.82	.61	.48
conference	H-Mean (ra2-M2)	n/a	.03	.05	.24
conference	H-Mean (ra2-M3)	n/a	.23	.3	.44
conference	H-Mean (rar2-M1)	n/a	.82	.63	.51
conference	H-Mean (rar2-M2)	n/a	.03	.05	.22
conference	H-Mean (rar2-M3)	n/a	.23	.31	.46
multifarm	H-Mean (diff)	00:00:14	.01	.01	.01
multifarm	H-Mean (same)	00:00:14	.20	.11	.08
largebio	FMA-NCI (small)	00:00:17	.964	.482	.321
largebio	FMA-NCI (whole)	00:03:31	.798	.443	.307
largebio	FMA-SNOMED (small)	00:00:36	.98	.226	.128
largebio	FMA-SNOMED (whole)	00:06:53	.933	.224	.127
largebio	SNOMED-NCI (small)	00:03:41	.967	.418	.267
largebio	SNOMED-NCI (whole)	00:07:16	.894	.408	.265

Table 2: RSDL Workbench Results for OAEI 2015

better for the tasks with same ontologies in different languages (same), because their hierarchies are identical. This is in contrast to the tasks with different ontologies (diff), where the ontologies have also different hierarchies. This explains the better quality of the produced alignments for the tasks with same ontologies in different languages.

largebio The data set of this track comprises the large biomedical ontologies Foundational Model of Anatomy (FMA), SNOMED CT, and NCI. These ontologies are semantically rich and contain a huge amount of concepts. *Largebio* consists of six test cases over three input ontologies. For each ontology pair, there are two tasks where whole ontologies (whole) or smaller fragments (small) are matched.

RSDLWB completed all the test cases in the given time frame of 10 hours. This is opposed to the OAEI 2014 campaign, when only the smaller FMA-NCI test could be completed [7]. In addition, the runtime was significantly improved. RSDLWB and LogMapLite [5] were the fastest systems altogether. Furthermore, RSDLWB and LogMapC [5] were the best systems in terms of precision across all test cases. In regard to the FMA-NCI test case, RSDLWB improved F-measure by 0.102.

2.1 Discussions on the way to improve the proposed system

As explained above, we plan to introduce features that exploit background knowledge in order to find non-trivial correspondences. It is also planned to use multilingual background knowledge from auxiliary ontologies like DBpedia to translate labels into different languages. This would enable support for the *multifarm* track.

Until now, the RSDLWB's classifiers were trained exclusively on the *benchmark* corpora. The integration of further classifiers that were trained on the other corpora might improve the results in regard to the different OAEI tracks.

The evaluation showed that RSDLWB's accuracy for class matching is much better than for property matching. One idea to improve the accuracy of property matching is to factor the similarity of their owning classes.

As explained in Sect. 1.2, the creation of a complete similarity matrix was replaced by heuristics to select seed pairs of concepts. Apparently, the fact that the matcher does not consider all concept pairs facilitates low recall. At the moment, the heuristic is too restrictive and only allows finding trivial correspondences with identical normalized labels. In the future, we want to explore further mapping candidates starting from the seed pairs.

3 Conclusion

The OAEI 2015 campaign showed a significant improvement of RSDLWB's runtime performance. This improvement was achieved by heuristics to select concepts pairs that are likely to match. The better runtime has enabled to complete all test cases of the *largebio* track, which is opposed to last year's OAEI campaign, when only one of six test cases could be completed. RSDLWB is one of the best systems in the OAEI 2015 campaign regarding the runtime. In general, RSDLWB has high precision when matching classes, but can be improved in regard to the matching of properties. In the future, we would to further improve RSDLWB's performance regarding recall.

Acknowledgments Special thanks to Henning Wachsmuth (Bauhaus-Universität Weimar) and Stefan Heindorf (University of Paderborn) for their support regarding machine learning.

References

1. Breiman, L.: Random Forests. Machine learning 45(1), 5–32 (2001)
2. Engels, G., Güldali, B., Soltenborn, C., Wehrheim, H.: Assuring Consistency of Business Process Models and Web Services Using Visual Contracts. In: Schürr, A., Nagl, M., Zündorf, A. (eds.) AGTIVE, LNCS, vol. 5088, pp. 17–31. Springer (2007)
3. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA Data Mining Software: An Update. SIGKDD Explor. Newsl. 11(1), 10–18 (Nov 2009)
4. Huma, Z., Gerth, C., Engels, G., Juwig, O.: Towards an Automatic Service Discovery for UML-based Rich Service Descriptions. In: France, R., Kazmeier, J., Breu, R., Atkinson, C. (eds.) MODELS. LNCS, vol. 7590, pp. 709–725. Springer (2012)
5. Jiménez-Ruiz, E., Grau, B.C.: LogMap: Logic-Based and Scalable Ontology Matching. In: 10th International Semantic Web Conference (ISWC). pp. 273–288 (2011)
6. Kullback, S., Leibler, R.A.: On Information and Sufficiency. The annals of mathematical statistics pp. 79–86 (1951)
7. Schwichtenberg, S., Gerth, C., Engels, G.: RSDL Workbench Results for 2014. In: Proceedings of the 9th International Workshop on Ontology Matching collocated with the 13th International Semantic Web Conference (ISWC). pp. 155–162 (2014)
8. Schwichtenberg, S., Gerth, C., Huma, Z., Engels, G.: Normalizing Heterogeneous Service Description Models with Generated QVT Transformations. In: Cabot, J., Rubin, J. (eds.) Modelling Foundations and Applications, LNCS, vol. 8569, pp. 180–195. Springer (2014)

ServOMBI at OAEI 2015

Nouha Kheder, Gayo Diallo

Univ. Bordeaux, ERIAS - Centre INSERM U1219, F-33000,
Bordeaux, France
first.last@u-bordeaux.fr

Abstract

We describe in this paper the ServOMBI system and the results achieved during the 2015 edition of the Ontology Alignment Evaluation Initiative. ServOMBI reuse components from the ServOMap ontology matching system, which uses to participate in the OAEI campaign, and implements new features. This is the first participation of the ServOMBI in the OAEI challenge.

1 Presentation of the System

ServOMBI (ServO based Mapping with Binary Indexing) is an ontology matching system [1] which is designed by reusing the overall workflow followed by the ServOMap large scale ontology matching system [2] grounded on top of the ServO Ontology Repository (OR) system [3]. ServOMap is able to handle ontologies which contain several hundred of thousands entities. To deal with large ontologies, the system relies on terminological indexing strategy provided by the ServO OR to reduce the search space and computes an initial set of candidate mappings based on the terminological description of the entities of the input ontologies.

With ServOMBI, new components and variant algorithms have been introduced in this new version in regards to the ServOMap system. Among these new features we have : –

- a binary indexing strategy to complement the terminological indexing of ServOMap for optimizing ontology navigation,
- a modified contextual similarity computation thanks to the introduction of the binary indexing strategy during the Machine Learning (ML) step,
- a new ML algorithm during the contextual similarity
- the introduction of parallelization of some tasks for optimization purposes
- the selection of the final mappings using a variant of a stable marriage problem algorithm [8]

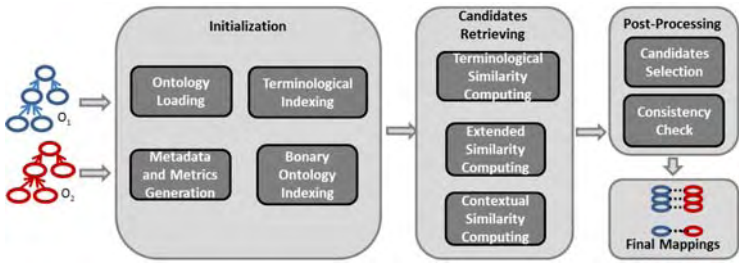


Figure 1: Overall process of ServOMBI.

In the 2015 edition, ServOMBI participated in the entity level matching tasks apart from the Multifarm task. In the following sections, we described the main characteristics of the system and the overall results obtained during this year edition of OAEI.

1.1 State, Purpose, General Statement

ServOMBI has been built on the basis of the ServOMap system. ServOMap is designed with the purpose of facilitating interoperability between different systems which are based on heterogeneous knowledge organization systems (KOS). The heterogeneity of these KOS may have several causes ranging from the language format they use to the level of formalism of the terminology which describe the entities they involve. Our system relies on Information Retrieval (IR) techniques [4] and a dynamic description of entities of different KOS for computing the similarity between them.

ServOMBI implements new features and strategies and reuse some components of the ServOMap system.

1.2 Specific techniques used

The overall process followed by the ServOMBI system is depicted on Figure 1. The initialization phase is modified by introducing the Binary Ontology Indexing (BOI).

1.2.1 Initialization phase

1. **Ontology Loading:** ServOMBI following the ServOMap approach relies on IR techniques for ontology matching. Each ontology to process is seen as a *corpus of semantic documents* to process. Each entity of the ontology is a document in the sense of IR. It is therefore necessary to identify the useful descriptors for indexing entities. The loading step perform the task of generating documents from entities. ServOMBI uses different reasoners (ELK [9], Hermit[11]) according to the size of the ontology to process.

2. **Metadata and Metrics Generation:** This step reuse the component implement in the ServOMap system and and identify 4 categories of matching tasks that are used to classify the input ontologies that are being processed : entity matching task (small, medium, big) and instances matching task.

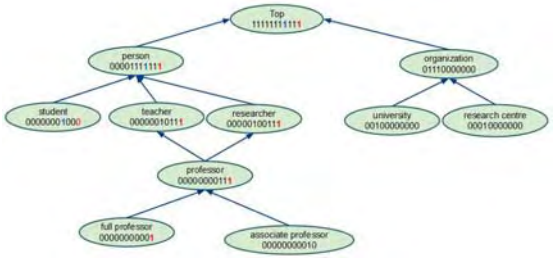


Figure 2: Example of binary indexing.

3. **Terminological Indexing:** Following a generic metamodel provided by the ServO OR, a terminological based inverted index is built from the documents generated during the loading step. ServOMBI introduces a tasks parallelization using multithreading as the input ontologies terminology indexing could be done separately.

4. **Binary Ontology Indexing:** To complement the terminological indexing and to optimize the performance in terms of processing times due to the contextual similarity computation (surrounding concepts lookups), we have introduced a binary indexing strategy for the input ontologies. This technique of taxonomical representation consists of representing each concept of the ontology by a binary code, is inspired by the CEDAR system [5]. A binary code is a number of n bits, with n the number of concepts within the processed ontology. Thus, each concept has a code (a bit vector) carrying a "1" in the position corresponding to his index and the index of any other elements that it subsumes. These bit vectors must be encoded as the reflexive transitive closure of the "is-a" relation obtained from subsort declarations. Concepts are represented by a graph. Figure 2 gives an example of a binary representation of the extract of an ontology in the academic domain. The concept *Professor* is the ancestor of *Full Professor* and *Associate Professor*. Therefore, if *Full Professor* is coded as the binary code of 1 and *Associate Professor* as the binary code of 2, *Full Professor* is coded as the binary code of 3.

1.2.2 Candidate Retrieving phase

Three main steps are used during the candidate mappings retrieving phase: terminological, extended (general purpose knowledge background) and contextual based candidate retrieving. The terminological based candidate retrieving uses indexes previously built and the IR common vectorial model. The extended candidate retrieving uses WordNet [7] while the contextual based candidate retrieving exploits the structure of each input Ontology, and the set of candidates provided by the terminological based candidate retrieving, in a ML strategy for acquiring more candidates. The ML strategy is based on the Logistic Model Trees (LMT) [10] algorithm.

Task	Precision	Recall	F-Measure
Anatomy	0.963	0.617	0.752

Table 1: Results of ServOMBI for the Anatomy track.

1.2.3 Post-Processing phase

In this phase two main steps are performed: the selction of the final mappings and consistency checking. The selection of final mappings implement an algorithm of the stable marriage problem [8].

1.3 Adaptation made for the evaluation

ServOMBI use the Lucene Apache IR library. Lucene provides functionalities for indexing and searching textual documents. The actual version of the matching system is based on the version 4 while the uploaded version for OAEI is based on the version 3.6. Their index format is slightly different. We have implemented the initial interactive matching [12] in ServOMBI using an oracle by modifying the validation process of the candidate mappings . This is performed after each round of candidate retrieving.

1.4 Link to the system and parameters file

The wrapped SEALS client for ServOMBI version used for the OAEI 2015 edition is available at http://lesim.isped.u-bordeaux2.fr/servo/ServOMBI. The instructions for testing the tool is described in the tutorial dedicated to the SEALS client[1].

1.5 Link to the set of provided alignments

The results obtained by ServOMap during OAEI 2015 are available at http://lesim.isped.u-bordeaux2.fr/servo/ServOMBI/oaei2015.zip/.

2 Results

We summarize in this section the results obtained by ServOMBI during the 2015 edition of OAEI.

2.1 Anatomy

The Anatomy track consists of finding an alignment between the Adult Mouse Anatomy and a part of the NCI Thesaurus (describing the human anatomy). The results achieved by ServOMBI are summarized by Table 1.

2.2 Conference

The Conference track contains 16 ontologies from the same domain (conference organization). They have been developed within the OntoFarm project[2]. This

[1] http://oaei.ontologymatching.org/2015/tutorial/tutorialv4.pdf
[2] http://owl.vse.cz:8080/ontofarm/

R.A.M.	Precision	F0.5 Measure	F1 Measure	F2 Measure	Recall
ra1-M1	0.64	0.64	0.64	0.65	0.65
ra1-M2	0.29	0.27	0.24	0.21	0.2
ra1-M3	0.61	0.6	0.59	0.59	0.58
ra2-M1	0.6	0.6	0.59	0.58	0.58
ra2-M2	0.29	0.27	0.24	0.21	0.2
ra2-M3	0.57	0.56	0.55	0.54	0.53
rar2-M1	0.59	0.59	0.6	0.61	0.61
rar2-M2	0.29	0.27	0.24	0.21	0.2
rar2-M3	0.56	0.56	0.55	0.55	0.55

Table 2: Results of ServOMBI for the Conference track.

Task	Precision	Recall	F-Measure
FMA-NCI	0.97	0.806	0.88
FMA-SNOMED	0.96	0.664	0.785

Table 3: Initial results of ServOMBI for the Large Bio track.

year the different tools are evaluated using i) crisp reference alignments where the confidence values for all matches are 1.0, ii) the uncertain version of the reference alignment where confidence values reflect the degree of agreement of a group of twenty people on the validity of the match [6] and iii) logical reasoning using violations of consistency and conservativity principles [15] [16]. Various reference alignments and evaluation modalities (R.A.M.) are used to assess the performance of the tooms. Thus, ra1 is the original reference alignment of the Conference track, ra2 is entailed reference alignment generated as a transitive closure computed on the original reference alignment (ra1) and rar2 is violation free version of reference alignment. Three different modalities are provided for these reference alignments, M1, M2 and M3 which contain respectively only classes, only properties and classes and properties.

The results obtained by ServOMBI according to these different modalities on the crisp reference alignments where the confidence value is 1.0 are summarized on table 2. The value of β is respectively set to 0.5, 1 (harmonic measure) and 2.

2.3 Largebio

The Large BioMed track consists of finding alignments between the Foundational Model of Anatomy (FMA), SNOMED CT, and the National Cancer Institute Thesaurus (NCI). The results obtained by ServOMBI for the small fragments of FMA-NCI task and FMA-SNOMED ontologies are summarize in Table 3

2.4 Interactive track

This track aims at offering a systematic and automated evaluation of matching systems with user interaction to compare the quality of interactive matching approaches in terms of F-measure and number of required interactions. For the 2015 edition, the Conference, Anatomy and Largebio tracks dataset are used

Error rate	Precision	Recall	F-Measure
0.0	1.00	0.617	0.763
0.1	1.00	0.587	0.740
0.2	1.00	0.553	0.712
0.3	1.00	0.519	0.683

Table 4: Results of ServOMBI for the Interactive track on the Anatomy dataset.

Error rate	Precision	Recall	F-Measure
0.0	1.00	0.650	0.788
0.1	1.00	0.637	0.778
0.2	1.00	0.622	0.767
0.3	1.00	0.627	0.770

Table 5: Results of ServOMBI for the Interactive track on the Conference dataset.

for the evaluation. Moreover, this year a domain experts with variable error rates, respectively 0.1, 0.2 and 0.3 are considered in addition to the perferct emmulated user (oracle) with error rate 0.0. ServOMBI participated for the first year to this track. The interaction implemented currently in the system is mainly to allow the user validating the provided candidate mappings. Tables 4, 5 and 6 give respectively the results obtained by the system on the Anatomy, Conference and Largebio dataset for the Interactive track. We note that for the Largebio interactive track, the ServOMBI was only able to match the FMA-NCI small fragments and FMA-SNOMED small fragments.

Overall ServOMBI improved its performance when compared to the results obtained with the normal Anatomy, Conference and Largebio track. However, the system make a greater number of requests compared to the other participating systems in the Interactive track.

2.5 Ontology Alignment for Query Answering

This track does not follow the usual OAEI tasks for evaluating the performance of participating systems [14]. Precision and Recall are calculated with respect to the ability of the generated alignments to answer a set of queries in a ontology-based data access scenario where several ontologies exist. This track uses the Conference dataset for the evaluation with two reference alignments, the publicly available Conference track alignment (RA1) and the repaired one (RAR1). Table 7 summarizes the results of ServOMBI which succeed with 6 out of 18 queries.

Error rate	Precision	Recall	F-Measure
0.0	1.00	0.737	0.847
0.1	1.00	0.716	0.832
0.2	1.00	0.688	0.813
0.3	1.00	0.660	0.792

Table 6: Results of ServOMBI for the Interactive track on the Largebio dataset.

Task	Precision	Recall	F-Measure
OAQA RA1	0.222	0.222	0.222
OAQA RAR1	0.222	0.222	0.222

Table 7: Results of ServOMBI for the OAQA track.

3 General Comments

We have participated in the 2012 and 2013 edition with the ServOMap system which achieved overall good results. The performance of this system is very good in particular for the tasks involving large ontologies. The new features implemented within ServOMBI did not lead to overall improved performances according to the results of the ServOMap system as expected. The contextual similarity computation, which is performed iteratively, is very time consuming and did not improved the overall recall of the system. In addition, while there is a gain in terms of computation times with concepts lookups, the BOI does not impact the overall performance of the system in terms of times taken to perform the matching tasks.

4 Conclusion

We have described in this paper the main functionalities of the ServOMBI ontology matching system and the overall results obtained during the 2015 OAEI edition. ServOMBI introduces a binary indexing strategy to complement the usual terminological indexing strategy used by the ServOMap system. The system achieved performance lower than expected according to the introduced features for the contextual similarity coputation. However it succed improving the F-Measure whith the interaction strategy. ServOMap continues to be developed in parralel and now include graph-based visualization.

As of future work, we envision to investigate an improved integration of the binary indexing and the contextual similarity computing. In addition, we plan to use combine multiple learning algorithms to improve the candidate selection during the contextual similarity computing.

5 Acknowledgments

This work has been partly supported by the project DRUG-SAFE funded ANSM (Agence Nationale de la Sécurité du Médicament). We also thank the organizers of OAEI with providing test dataset and the evaluation infrastructure.

References

[1] Jerôme Euzenat and Pavel Shvaiko, "Ontology Matching", Springer-Verlag, Heidelberg, 2013

[2] Gayo, Diallo. An effective method of large scale ontology matching. Journal of Biomedical Semantics, vol:5(44), 2014. DOI:10.1186/2041-1480-5-44

[3] Gayo Diallo. Efficient building of local repository of distributed ontologies. IEEE Proc. of the 7th International Conference on Signal Image Technology & Internet Based Systems (SITIS'11). K Yetongnon, R Chbeir and A Dipanda eds. Nov 28- Dec 1st 2011, Dijon, France

[4] Ricardo Baeza-Yates, Berthier Ribeiro-Neto. Modern Information Retrieval—The Concepts and Technology behind Search. 2nd Edition, Pearson, 2011

[5] Samir Amir, Hassan Aït-Kaci. CEDAR: Efficient Reasoning for the Semantic Web. Proceedings of the 10th IEEE International Conference on Signal Image Technology & Internet-Based Systems (SITIS 2014), Marrakech, Morocco, 2014

[6] Michelle Cheatham, Pascal Hitzler. Conference v2.0: An Uncertain Version of the OAEI Conference Benchmark. International Semantic Web Conference (2) 2014: 33-48

[7] George A. Miller. Wordnet: A lexical database for english. Communications Of The ACM, 38:39–41, 1995.

[8] Iwama, Kazuo; Miyazaki, Shuichi. A Survey of the Stable Marriage Problem and Its Variants. pp. 131–136. doi:10.1109/ICKS.2008.7

[9] Yevgeny Kazakov, Markus Krötzsch, František Simančík. Unchain My EL Reasoner. In Riccardo Rosati, Sebastian Rudolph, Michael Zakharyaschev, eds.: Proceedings of the 24th International Workshop on Description Logics (DL-11). CEUR Workshop Proceedings 2011

[10] Niels Landwehr, Mark Hall, and Eibe Frank. Logistic Model Trees . In Machine Learning 59 (1-2) 161-205, 2005

[11] Birte Glimm, Ian Horrocks, Boris Motik , Giorgos Stoilos, Zhe Wang. HermiT: An OWL 2 Reasoner. Journal of Automated Reasoning. Volume 53, Issue 3, pp 245-269, 2014

[12] Heiko Paulheim, Sven Hertling, Dominique Ritze. "Towards Evaluating Interactive Ontology Matching Tools". ESWC 2013

[13] Gerard Salton. (1979). Mathematics and information retrieval. Journal of Documentation,35 (),–29

[14] Alessandro Solimando, Ernesto Jimenez-Ruiz, and Christoph Pinkel. Evaluating Ontology Alignment Systems in Query Answering Tasks. Poster paper at International Semantic Web Conference (ISWC). 2014

[15] Alessandro Solimando, Ernesto Jiménez-Ruiz, Giovanna Guerrini. Detecting and Correcting Conservativity Principle Violations in Ontology-to-Ontology Mappings. International Semantic Web Conference (2) 2014: 1-16.

[16] Alessandro Solimando, Ernesto Jiménez-Ruiz, Giovanna Guerrini. A Multi-strategy Approach for Detecting and Correcting Conservativity Principle Violations in Ontology Alignments. OWL: Experiences and Directions Workshop 2014 (OWLED 2014). 13-24

STRIM Results for OAEI 2015 Instance Matching Evaluation

Abderrahmane Khiat[1] , Moussa Benaissa[1] and Mohammed Amine Belfedhal[2]

[1] LITIO Laboratory, University of Oran1 Ahmed Ben Bella, Oran, Algeria
abderrahmane_khiat@yahoo.com , moussabenaissa@yahoo.fr
[2] Evolutionary Engineering and Distributed Information Systems Laboratory (EEDIS), Djillali LIabes University of Sidi Bel Abbes, Algeria
Mohammed.belfedhal@gmail.com

Abstract. The interest of instance matching grows everyday with the emergence of linked data. This task is very necessary to interlink semantically data together in order to be reused and shared. In this paper, we introduce STRIM, an automatic instance matching tool designed to identify the instances that describe the same real-world objects. The STRIM system participates for the first time at OAEI 2015 in order to be evaluated and tested. The results of the STRIM system on instance matching tracks are so far quite promising. In effect, the STRIM system is the top system on SPIMBENCH tracks.

Keywords: String-Based Similarity, Instance Mapping, Instance Matching, Linked Data, Web of Data, Semantic Interoperability, Semantic Web.

1 Introduction

The current Web, contains *documents* in various formats (PDF, Excel, HTML file, etc.) *connected* by *hypertext links*, also known as the *Web of Documents*. Note that, we mean by document, if the content is unstructured and not exploitable i.e. the semantic the content is not presented. Contrary to data, where the content is structured and exploitable i.e. the semantic of the content is presented using RDF for example.

The *inadequacy* of the *Web of Documents* resides in the fact that the *content* of *these documents* is probably *unstructured* and *its semantic is not presented* which means that it *is not exploitable* and *untreatable automatically* in different applications, either by the *machine* or by *expressive queries*.

In order to deal with these problems, and especially for the re-use and sharing of content, the *transition* from the *document* to the *data* is very necessary. This involves the *use of semantic web technologies* in order to (a) publish structured data on the Web, (b) make possible, the links between data from one data source to data within other data sources. These two points are very important to ensure *semantic interoperability*.

These data should be expressed using the RDF language (Resource Description Framework [see section 2.1]) to achieve the two major points that we have mentioned before in order to enable the semantic interoperability, which led to the emergence of the Web of Data. The data presented and structure in this form (RDF) can be easily interpreted by the computer, re-used in applications and easily linked with other data. If

the data are easily linked the computer can work through relationships with other data and in this case the interoperability will be ensured. Other advantages of Linked Data among others are: improving the data quality, less human intervention and processing and short development cycles (quicker and save time).

With the effort of Linked Data Community to publish existing open license datasets as Linked Data on the Web and interlink things between different data sources, the Web of Linked Data has seen remarkable increase over the past years. In terms of statistics, in 2007, over 500 million RDF triples published on the web with around 120,000 RDF links between data sources. In 2010, over the 28.5 billion triples, in 2011 over 31.6 billion triples and in 2013 over 50 billion triples. According to these statistics, the Linked Data seems to be increasing drastically [6].

Linked Data, by definition, links the instances of multiple sources. A common way to link these instances to others is to use the owl:sameAs property. The enormous volume of data already available on the web and its continuity to increase, requires techniques and tools capable to identify the instances that describe the same real-world objects automatically.

With the OAEI evaluation campaign which distinguishes between matching systems that have participated in the category of ontology matching and those that have participated in the category of instance matching, these tools can be tested and evaluated. However, few systems[3] [10] namely InsMT, LogMap and RiMOM-IM have participated to test their performance at instance matching track of OAEI 2014.

In this paper we deal with two challenges namely:

1. How to link the distributed and heterogeneous data which are described with instances.
2. How to deal with the huge volume of data available on the web and its continuity to increase [14].

Indeed, the Solution to this problem consists to provide techniques and tools capable to identify the instances that describe the same real-world objects automatically.

In this paper, we describe the STRIM system in order to resolve automatically the instance matching problem. The STRIM system, extracts first all information about the two instances to be matched and normalizes them using NLP techniques. Then, it applies edit distance as a matcher to calculate the similarities between the normalized information. Finally, the approach selects the equivalent instances based on the maximum of shared information between the two instances.

The STRIM system has participated for the first time at OAEI evaluation campaign and it provides very good results in terms of precision, recall and f-measure.

The rest of the paper is organized as follows. First, the preliminaries on instance matching are presented in section 2. In the Section 3, we presented the related work on instance matching systems that participated in Instance Matching Track of OAEI 2014. In the Section 4, we describe our system by giving a detailed account of our approach. The experimentation results is presented in Section 5. The Section 6 contains concluding remarks and sets directions for future work.

[3] The declaration of OAEI 2014 evaluation campaign about instance matching systems 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.

2 Preliminaries

In this section, we present the basic notions related to Instance Matching by explaining the linked data and instance matching definition.

2.1 Linked Data Principle

The Linked Data consist to relate data with typed links across the Web using URIs, HTTP and RDF. The linked Data principles are defined by Tim Berners-Lee in 2007 [11]. These principles are as follow:

- Use URIs as names for things.
- Use HTTP URIs so that people can look up those names.
- When someone looks up a URI, provide useful RDF information.
- Include RDF statements that link to other URIs so that they can discover related things.

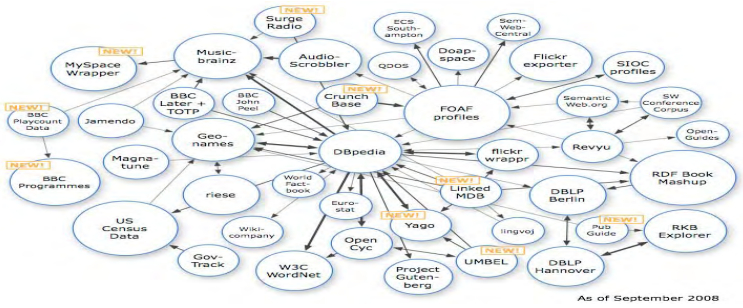


Fig. 1: Linked Data

The Linked Data (Fig. 1), by definition [12], links the instances of multiple sources. A common way to link the instances in these sources to others, is the use of the owl:sameAs property. Instance matching is required to interlink these data.

2.2 RDF Language

These data should be expressed using the RDF language (Resource Description Framework) to achieve the two major points that we have mentioned before in order to enable the semantic interoperability. The RDF language is a graph model to formally describe Web resources and metadata, in order to allow automatic processing of such descriptions [13][1][2]. An RDF file thus formed is a labeled directed multi-graph. Each triplet corresponds to a directed arc whose label is the predicate, the source node is the subject and the target node is the object.

2.3 Instance Matching Definition

The Instance Matching (Fig.2) is a process that starts from collections of data as input and produces a set of mappings (simple or complex) between entities of the collections as output [5].

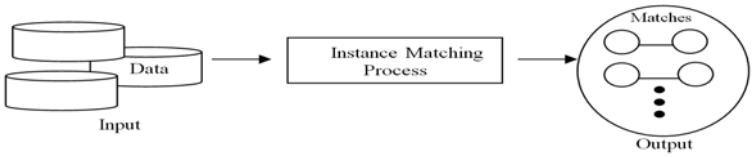


Fig. 2: Instance Matching Process

2.4 Entity Resolution Notion

Definition [5]: Let D1 and D2 be represent two datasets, each one contains a set of data individuals T_i which are structured according to a schema O_i. Each individual $I_i j$ T_i describes some entity w_j.

Two individuals are said to be equivalent I_j I_k if they describe the same entity w_j = w_k according to a chosen identity criterion. The goal of the entity resolution task is to discover all pairs of individuals ($I_1 i$, $I_2 j$) — $I_1 i$ T_1, $I_2 j$ T_2 such that $w_1 i = w_2 j$.

In the context of linked data, datasets D_i are represented by RDF graphs. Individuals I_i T_i are identified by URIs and described using the classification schema and properties defined in the corresponding ontology O_i.

Example of Instance Matching We give below an example that shows how to link data from DBpedia with other data sources using the owl:sameAs property.

<http://dbpedia.org/resource/Berlin>
owl:sameAs
<http://sws.geonames.org/2950159>

3 Related Work

We present and discuss in this section the major works relevant to instance matching that participated at OAEI 2014 evaluation campaign. Only two systems succeed to finish all sub-tracks of instance matching track of OAEI 2014, namely RiMOM-IM and our previous InsMT system. We cite in exhaustive way only the instance matching systems that have participated in OAEI 2014 evaluation campaign and which are the object of comparison with our system STRIM.

1) LogMap [7]: The LogMap family participated with four different versions namely LogMap, LogMap-Bio, LogMap-C and LogMapLite in OAEI 2014. Only two versions (LogMap and LogMap-C) of them have participated at instance matching track. The LogMap-family system is a highly scalable ontology matching system with built-in reasoning and inconsistency repair capabilities. The two versions of LogMap systems identifies mappings between instances. The LogMap and LogMap-C systems finish only the first sub-track of instance matching of OAEI 2014 which is Identity Recognition.

2) RiMOM-IM [9] [3] [4]: is an acronym of Risk Minimization based Ontology Mapping Instance Matching. The principle of RiMOM-IM is to construct a document from the dataset by extracting the instances information. Then, it uses cosine-similarity to compare documents. The version of RiMOM-IM system that participated in OAEI 2014 for instance matching is developed based on ontology matching system RiMOM with some changes in objective. The objective of RiMoM-IM is to solve the challenges in large-scale instance matching by proposing a novel blocking method.

3) InsMT(L) [8]: is an acronym of Instance Matching at Terminological (Linguistic) level. InsMT(L) has participated for the first time in OAEI 2014. The principle of InsMT(L) is to use String-based algorithms (and WordNet as matcher at linguistic level) in order to calculate similarities between instances after the annotation step. The similarities calculated by each matcher are aggregated using the average aggregation strategy after a local filtering. Finally InsMT(L) system operates a global filtering in order to identify the alignment. The InsMT(L) system shows good results in terms of recall on different sub-tracks of instance matching of OAEI 2014. The InsMT(L) system finishes all sub-tracks of instance matching of OAEI 2014 which is Identity Recognition and Similarity Recognition.

4) Other Approaches:

There are several other instance matching approaches like HMatch [18], FBEM [17], SILK [16] and the works proposed in [15] which are not covered by this paper due to minor importance for our approach. These instance matching approaches have not participated in instance matching track of OAEI 2014. With respect to these approaches, we did not take them in consideration because we do not have their official results for the experimental protocol of OAEI in 2014.

As we have mentioned before, with the high number of publications about interlinking approaches only a few systems have participated at OAEI 2014. These systems are LogMap, RiMoM-IM and our previous InsMT(L) system.

4 STRIM: STRing based algorithm for Instance Matching

We summarize the process of our approach to provide a general idea of the proposed solution. It consists in the following successive phases:

4.1 Extraction and Normalization

The system extracts from each individual Ii P_1 m_1; P_2 m_2,... a set of information m_1, m_2, ... using different properties P_1, P_2, ... Then, NLP techniques are applied to normalize these infrmation. In particular, three pre-processing steps are performed: (1)

case conversion (conversion of all words in same upper or lower case) (2) lemmatization stemming and (3) stop word elimination. Since String based algorithm is used to calculate the similarities between information, these steps are necessary.

4.2 Similarity Calculation

In this step, the system calculates the similarities between the normalized informations using edit distance as string matcher. Our system selects the maximum similarity values calculated between different informations by edit distance. If two informations are the same (based on maximim similarity values) the counter is incremented to 1, etc.

4.3 Identification

Finally, we apply a filter on maximum counter values in order to select the correspondences which mean that the selected correspondences (equivalent individuals) are those who share maximum informations.

5 Experimentation

In this section, we present the results (Tab. 1) obtained by running our STRIM system on instance matching tracks of OAEI 2015 evaluation campaign.

Table 1: The Results of STRIM System

System	Track	Precision	F-measure	Recall
STRIM	sandbox val-sem task	0.90	0.95	0.99
LogMap	sandbox val-sem task	0.99	0.92	0.86
STRIM	mainbox val-sem task	0.91	0.95	0.99
LogMap	mainbox val-sem task	0.99	0.92	0.85
STRIM	sandbox val-struct task	0.99	0.99	0.99
LogMap	sandbox val-struct task	0.99	0.90	0.82
STRIM	mainbox val-struct task	0.99	0.99	0.99
LogMap	mainbox val-struct task	0.99	0.90	0.82
STRIM	sandbox val-struct-sem task	0.91	0.95	0.99
LogMap	sandbox val-struct-sem task	0.99	0.88	0.79
STRIM	mainbox val-struct-sem task	0.91	0.95	0.99
LogMap	mainbox val-struct-sem task	0.99	0.88	0.79

Only two systems have participated at SPIMBNNCH tracks namely the LogMap and STRIM systems. The SPIMBENCH consists of the following three different tasks: val-sem, val-struct and val-sem-struct. Each task has two tests (1) the Sandbox which contains two datasets in small scale and (2) the Mainbox which contains two datasets in

large scale. The goal of three tasks consists to determine when two OWL instances describe the same Creative Work. However, the three tasks have been produced by altering a set of original data. In other words, the datasets of the val-sem task have been produced by using value-based and semantics-aware transformations. For the datasets of the val-struct task have been produced by using value-based and structure-based transformations. Finally the datasets of the val-sem-struct task have been produced by using value-based, structure-based and semantics-aware transformations.

We have evaluated the results of STRIM system based on the results obtained on Mainbox tests. The reason is that these tests were blind (i.e. the reference alignment is not given to the participants) during the evaluation of Instance matching systems by the OAEI evaluation campaign. On the other side, in the Sandbox tests, the reference alignment were available to help the instance matching systems to configure theirs parameters.

Regarding F-measure results, the STRIM system seems to achieve the best results before the LogMap system. The F-measure is always more than 95%. we can remark that STRIM system achieve high recall for the three tasks. It always equal to 99%.

* As conclusion, the result proves that our STRIM system is effective and efficient for the three tasks of SPIMBENCH track of OAEI 2015.

6 Conclusion

In this article, we have introduced STRIM, an instance matching system which consists in identifying the instances that describe the same real-world objects automatically. Our approach is useful, especially when the instances contain terminological information.

The STRIM system is composed of three steps: the first step consists in extracting and normalizing all information about the two instances to be matched. The second step consists in applying an edit distance as a matcher to calculate the similarities between the normalized information. The final step, consists in selecting the equivalent instances based on the maximum of shared information between the two instances.

The STRIM system has participated for the first time at OAEI evaluation campaign and it provides very good results in terms of precision, f-measure and recall at Instance Matching of OAEI 2015.

As future perspective, we attempt to apply STRIM to link data on could computing environment and develop other approaches.

References

1. J. Euzenat and P. Shvaiko Ontology Matching, Second Edition, Springer-Verlag, Heidelberg, pp. 1-511, 2013.
2. M. Ehrig Ontology Alignment: Bridging the Semantic Gap, Semantic Web And Beyond Computing for Human Experience 4, Springer, pp. 1-250, 2007.
3. Z. Wang, X. Zhang, L. Hou,Y. Zhao, J. Li, Y. Qi and J. Tang RiMOM results for OAEI 2010, In The Proceedings of the 4th International Workshop on Ontology Matching co-located with the 9th International Semantic Web Conference (ISWC 2010), pp. 195-202. CEUR-WS.org, Vol. 689, Shanghai, China, 2010.

4. J. Li, J. Tang, Y. Li and Q. Luo RiMoM: a Dynamic Multistrategy Ontology Alignment Framework, Journal IEEE Transactions on Knowledge and Data Engineering, vol. 21, No. 8, pp. 1218-1232, 2009.
5. A. Ferrara, A. Nikolov, J. Noessner and F. Scharffe Evaluation of instance matching tools: The experience of OAEI, Journal of Web Semantics: Science, Services and Agents on the World Wide Web, Vol. 21 pp. 49-60, 2013.
6. C. Bizer, T. Heath, and T. Berners-Lee. Linked Data - The Story So Far. International Journal on Semantic Web and Information Systems (IJSWIS), 2009.
7. E. Jimnez-Ruiz, B. C. Grau, W Xia, A. Solimando, X. Chen, V. Cross, Y. Gong, S. Zhang and A. Chennai-Thiagarajan LogMap family results for OAEI 2014. In Proceedings of the 9th International Workshop on Ontology Matching co-located with the 13th International Semantic Web Conference (ISWC 2014), October 20, pp. 126-134. CEUR-WS.org, Trentino, Italy, 2014.
8. A. Khiat, M. Benaissa, InsMT / InsMTL results for OAEI 2014 instance matching. In Proceedings of the 9th International Workshop on Ontology Matching co-located with the 13th International Semantic Web Conference (ISWC 2014), October 20, pp. 120-125. CEUR-WS.org, Trentino, Italy, 2014.
9. C. Shao, L. Hu and J. Li, RiMOM-IM results for OAEI 2014. In Proceedings of the 9th International Workshop on Ontology Matching co-located with the 13th International Semantic Web Conference (ISWC 2014), October 20, pp. 149-154. CEUR-WS.org, Trentino, Italy, 2014.
10. Z. Dragisic, K. Eckert, J. Euzenat, D. Faria, A. Ferrara, R. Granada, V. Ivanova, E. Jimnez-Ruiz, A. O. Kempf, P. Lambrix, S. Montanelli, H. Paulheim, D. Ritze, P. Shvaiko, A. Solimando, C. Trojahn, O. Zamazal, B. C. Grau, Results of the Ontology Alignment Evaluation Initiative 2014. In Proceedings of the 9th International Workshop on Ontology Matching co-located with the 13th International Semantic Web Conference, pp. 61-104. CEUR-WS.org, Trentino, Italy, 2014.
11. Tim Berners-Lee. Linked Data - Design Issues, 2006. http://www.w3.org/DesignIssues/LinkedData.html. 7, 26, 82.
12. R. Parundekar, C.A. Knoblock, J.L. Ambite, Linking and building ontologies of linked data. In: Proceedings of the 9th International Semantic Web Conference (ISWC 2010). Shanghai, China, 2010.
13. G. Klyne and J. J. Carroll. Resource Description Framework (RDF): Concepts and Abstract Syntax - W3C Recommendation, http://www.w3.org/TR/rdf-concepts/, 2004.
14. P. Shvaiko and J. Euzenat. Ten challenges for ontology matching. In R. Meersman and Z. Tari, editors, On the Move to Meaningful Internet Systems: OTM 2008, volume 5332 of Lecture Notes in Computer Science, pp. 11641182. 2008.
15. D. Engmann and S. Mamann. Instance matching with COMA++. In Proceedings of Datenbanksysteme in Business, Technologie and Web(BTW 07), pages 2837, 2007.
16. J. Volz, C. Bizer, M. Gaedke, and G. Kobilarov. Discovering and maintaining links on the web of data, In The Proceedings of 8th International Semantic Web Conference (ISWC 2009), A. Bernstein, D. Karger, T. Heath, L. Feigenbaum, D. Maynard, E. Motta, and K. Thirunarayan, editors, The Semantic Web - ISWC 2009, volume 5823 of Lecture Notes in Computer Science, pp. 650665. Springer Berlin / Heidelberg, 2009.
17. H. Stoermer and N. Rassadko. Results of okkam feature based entity matching algorithm for instance matching contest of oaei 2009, 2009.
18. S. Castano, A. Ferrara, S. Montanelli, and D. Lorusso. Instance matching for ontology population. In Proceedings of the 16th Italian Symposium on Advanced Database Systems, pages 121132, 2008.

XMap : Results for OAEI 2015

Warith Eddine DJEDDI[a,b], Mohamed Tarek KHADIR[a] and Sadok BEN YAHIA[b]

[a]LabGED, Computer Science Department, University Badji Mokhtar, Annaba, Algeria
[b]Faculty of Sciences of Tunis, University of Tunis El-Manar, LIPAH-LR 11ES14, 2092, Tunisia
{djeddi,khadir}@labged.net
sadok.benyahia@fst.rnu.tn

Abstract. This paper describes the configuration of XMap for the OAEI 2015 competition and discusses its results. XMap is able to automatically adapt to the matching task, choosing the best conguration for the given pair of ontologies. This is our third participation in the OAEI and we can see an overall improvement on nearly every task.

1 State, purpose, general statement

XMap [1] [2] is a highly scalable ontology matching system, which is able to deal with hundreds of thousands of entities with an efficient computation time [3]. It is a fast and effective high precision system able to perform matching large ontologies. A semantic similarity measure has been defined using UMLS and WordNet [4] to provide a synonymy degree between two entities from different ontologies, by exploring both their lexical and structural context. XMap relies on the Microsoft Translate API to translate ontologies into many languages.

1.1 Specific techniques used

A high-level view of mapping process is depicted in Figure 1. It is a multi-layer system which uses three different layers to perform the ontology alignment process: a terminological layer, a structural layer and an alignment layer. The output values of each layer serves as input to the upper one and each layer provides an improvement in the computation of the similarity between concepts. Figure 1 shows the architecture of the XMap system.

Matchers in XMap are the algorithms that compare two ontologies and return an alignment between them. The matchers employed various strategies (entity label, structural description of concepts, range for relations, instantiated attributes or extensional descriptions) in each layer which are listed below:

a) Terminological Layer

The terminological layer is responsible for carrying out the process of computing the similarity between the entity names within the ontologies, combining linguistic similarity with the semantic elements of the context of the entities. This layer receives as inputs the values of the string similarity, the linguistic similarity, the semantic similarity and translation-based similarity computed within the lexical-semantic module. The output variable represents the terminological similarity:

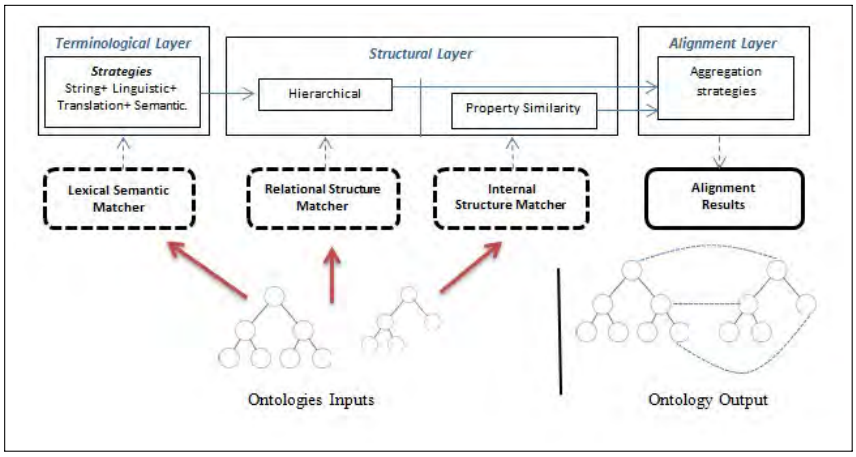


Fig. 1. Architecture for XMAP.

1. The string strategies usually can be applied to the name, the label or the comments concerning entities to discover those which are similar. In general, it can be used for comparing class names and/or URIs. The scaled range is [0, 1] for comparing strings. Our system applies many terminological approaches for computing the similarity measures between two terms: the Levenshtein distance, the Jaro Winkler distance, the n-grams, the Jaccard distance, the Cosine, etc. Note that XMap does not currently store or use comments, definitions, nor instances;
2. The linguistic strategies explore the semantic similarity of the concepts and relations labels. The linguistic based matchers use the external resources WordNet and UMLS to find the semantic similarity between two entities;
3. The translation-based strategies use an automatic translation for obtaining correct matching pairs in multilingual ontology matching. The translation is done locally by querying Microsoft Translator for the full name;
4. The semantic strategies based on auxiliary sources use a domain knowledge available from external sources, such as WordNet, to find additional information for the concepts (synonyms) and the relationships between them. The semantic similarity is incorporated with the aim of adding context information of the concepts during the mapping process.

b) Structural Layer

The structural layer performs two key tasks related to the structure of ontologies. One is the computation of the similarity between the concepts taking into account the taxonomic hierarchy, as well as the computation of the similarity using the information of the internal structure of concepts, i.e., their properties, types and cardinality restrictions:

1. Structural strategies are usually based on the internal structure of an entity or its relations to other entities as a source of detecting correspondences. The first using the relational structure of concepts in the ontology, specifically the taxonomic hierarchy, and the second using the information of the internal structure of concepts, including their properties, types and cardinality restrictions;

2. Constraint strategies consider the concepts and properties data types and cardinalities. They are usually used to provide supplementary information, not as primary matchers (i.e., sting matcher or linguistic matcher); these techniques consider criteria regarding the internal structure of the entities, such as the domain and range of the properties or the types of the attributes, to calculate the similarity between them.

c) Alignment Layer

The alignment layer is the final layer and its aim is to provide the final similarity matrix between the concepts taking into account the influence of the number of properties and the value of similarity that properties bring to the final similarity between them. Once the similarity between ontology entities are available based on different strategies (e.g., string similarity, semantic similarity, structural similarity), aggregating similarities algorithms are needed to combine matchers. Combining and filtering the similarity values obtained from the different matchers, comes most often, to combine similarity values using three types of aggregation operator; these strategies are aggregation, selection and combination [2]. Furthermore, those pairs of concepts with similarity values equal to or greater than a particular threshold are retained in order to obtain the mapping suggestions.
For the requirements of different ontology matching tasks, the selected alignment in XMap can be one to one, one to many, or many to many alignments. Whereas in our case, the desired cardinality in ontology matching is typically one-to-one. The matching rules are created via the Java API Alignment Format, allowing the generation of outputs in different formats.

2 Results

In this section, we present the evaluation results obtained by running XMap under the SEALS client with *Benchmark*, *Anatomy*, *Conference*, *Multifarm* and *Large Biomedical Ontologies* tracks. Adding to that, we present the results of the test *Ontology Alignment for Query Answering* which not follow the classical ontology alignment evaluation on the SEALS platform.

Benchmark XMap performs very well in terms of Precision (1.0) while flagging out a low recall (0.4) in the Benchmark track. Those low values are explained by the fact that ontological entities with scrambled labels and lexical similarity become ineffective. Whereas for the others two test suites our algorithm performed worse in terms of F-Measure because our system does not handle ontology instances. Table 1 summarises the average results obtained by XMap.

Table 1. Results for Benchmark track.

Test	Precision	Recall	F-Measure
biblio	1.0	0.40	0.57
energy	1.0	0.22	0.51

Anatomy The Anatomy track consists of finding an alignment between the Adult Mouse Anatomy (2744 classes) and a part of the NCI Thesaurus (3304 classes) describing the human anatomy. XMap achieves a good F-Measure value of ≈89% in a reasonable amount of time (50 sec.) (see Table 2). In terms of F-Measure/runtime, XMap is ranked 2nd among the 15 tools participated in this track.

Table 2. Results for Anatomy track.

System	Precision	F-Measure	Recall	Time(s)
XMap	0.928	0.896	0.865	50

Conference The Conference track uses a collection of 16 ontologies from the domain of academic conferences. Most ontologies were equipped with OWL DL axioms of various types; this opens a useful way to test our semantic matchers. The match quality was evaluated against the original (ra1) as well as entailed reference alignment (ra2) and violation free version of reference alignment (ra2). As Table 3 shows, for the three evaluations, we achieved a good F-Measure values.

For each reference alignment, three evaluation modalities are applied : a) M1 only contains classes, b) M2 only contains properties, c) M3 contains classes and properties.

Table 3. Results for Conference track.

	Precision	F-Measure 1	Recall
Original reference alignment (ra1)			
ra1-M1	0.86	0.73	0.63
ra1-M2	0.67	0.22	0.13
ra1-M3	0.85	0.68	0.56
Entailed reference alignment (ra2)			
ra2-M1	0.81	0.68	0.58
ra2-M2	0.78	0.25	0.15
ra2-M3	0.81	0.63	0.51
Violation reference alignment (rar2)			
rar2-M1	0.8	0.69	0.62
rar2-M2	0.78	0.27	0.16
rar2-M3	0.8	0.64	0.54

Multifarm This track is based on the translation of the OntoFarm collection of ontologies into 9 different languages. XMap's results are showed in Table 4.

Table 4. Results for Multifarm track.

System	Different ontologies			Same ontologies		
	P	F	R	P	F	R
XMap	0.22	0.24	0.27	0.66	0.37	0.27

Large biomedical ontologies This track consists of finding alignments between the Foundational Model of Anatomy (FMA), SNOMED CT, and the National Cancer Institute Thesaurus (NCI). There are 5 sub-tasks corresponding to different sizes of input ontologies (small fragments and whole ontology for FMA and NCI and small and large fragments for SNOMED CT). 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 desactivated. The results obtained by XMAP-BK are depicted by Table 6. XMAP-BK provided the best results

Table 5. Results for the Large BioMed track.

Test set	Precision	Recall	F-Measure	Time(s)
Small FMA-NCI	0.971	0.902	0.935	31
Whole FMA-NCI	0.872	0.849	0.860	337
Small FMA-SNOMED	0.968	0.847	0.903	49
Whole FMA- Large SNOMED	0.769	0.844	0.805	782
Small SNOMED-NCI	0.928	0.606	0.733	396
Whole NCI- Large SNOMED	0.913	0.536	0.675	925

(ranked 1st) among the 12 participating systems in terms of F-measure in FMA-NCI and FMA-SNOMED matching sub-tasks. In general, we can conclude that XMap achieved a good precision/recall values. The high recall value can be explained by the fact that UMLS thesaurus contains definitions of highly technical medical terms.

Ontology Alignment for Query Answering The objective of this test is to check the ability of the generated alignments to answer a set of queries in an ontology-based data access scenario where several ontologies exist. Table 6 shows the F-measure results for the whole set of queries. XMap was one of the 5 matchers whose alignments allowed to answer all the queries of the evaluation.

Table 6. Results for Ontology Alignment for Query Answering.

System	RA1 Reference			RAR1 Reference		
	P	R	F	P	R	F
XMap	0.778	0.675	0.702	0.720	0.654	0.671

3 General comments

3.1 Comments on the results

This is the third time that we participate in the OAEI campaign. We foresee an improvement in the performance of our system which consists of expanding the supported domain of matching problems, such that large-scale biomedical or multilingual ontologies can be matched as well. The official results of OAEI 2015 show that XMap is competitive with other well-known ontology matching systems in all OAEI tracks. The current version of XMap has shown a significant improvement both in terms of matching quality and runtime. Additionally, to improve our f-measure for large biomedical ontologies we made use of the UMLS Metathesaurus. Finally, we pre-compiling a local dictionary in order to avoid multiple accesses to the Microsoft Translator during the matching process.

3.2 Comments on the OAEI 2015 procedure

As a third participation, we found the OAEI procedure very convenient and the organizers very supportive. The OAEI test cases are various, and this leads to a comparison on different levels of difficulty, which is very interesting. We found that SEALS platform is a precious tool to compare the performance of our system with the others.

4 Conclusion

In this paper, we presented the results achieved during the 2015 edition of the OAEI campaign. The system managed to improve its performance significantly compared to the previous year, which is reflected in the performance of the different tracks. We have used the UMLS resource for better discarding incorrect mappings for life sciences related ontologies. Moreover, we implemented a cross-lingual ontology matching approach in order to align ontologies in different languages.

References

1. Djeddi, W., Khadir, M. T.: A Novel Approach Using Context-Based Measure for Matching Large Scale Ontologies. In Proceedings of 16th International Conference on Data Warehousing and Knowledge Discovery (DAWAK 2014), September 2-4, pp. 320–331. Springer, Munich, Germany (2014)
2. Djeddi, W., Khadir, M.T.: Ontology alignment using artificial neural network for large-scale ontologies. In the International Journal of Metadata, Semantics and Ontologies (IJMSO), Vol.8, No.1, pp.75-92 (2013)
3. Dragisic Z., Eckert K., Euzenat J., Faria D., Ferrara A., Granada R., Ivanova V., & al.: 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, 61–104, (2014).
4. Fellbaum, C. : WordNet: An Electronic Lexical Database, MIT Press, Cambridge, MA (1998)

Instance-Based Property Matching in Linked Open Data Environment

Cheng Xie[1], Dominique Ritze[2], Blerina Spahiu[3], and Hongming Cai[1]

[1] Shanghai Jiao Tong University
[2] University of Mannheim
[3] University of Milano-Bicocca
chengxie@sjtu.edu.cn

Abstract. Instance matching frameworks that identify links between instances, expressed as `owl:sameAs` assertions, have achieved a high performance while the performance of property matching lags behind. In this paper, we leverage `owl:sameAs` links and show how these links can help for property matching.

Keywords: Property matching, Instance-based matching, Linked Open Data

Introduction The performance of ontology matching systems on property matching lags significantly behind that on class and instance matching [1]. Current state-of-the-art techniques achieve a high performance on instance matching which focus on finding `owl:sameAs` links between LOD datasets [2]. These linked instances give an important information to the property matching process which we further explore in this paper. We argue that `owl:sameAs` instance pairs share similar values on similar properties. For this issue, we investigate to which extent we can automatically find matching properties by exploiting `owl:sameAs` instance pairs.

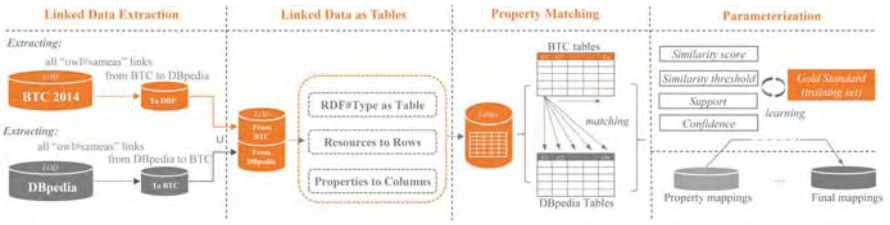


Fig. 1. General matching pipline

Approach Figure1 shows a concrete example of matching DBpedia to BTC2014[4]. The proposed approach has four steps which are described in detail below.

Linked Data Extraction: As first step, we extract all `owl:sameAs` triples whose subject is an instance in BTC2014 while the object is an instance in DBpedia. With the same heuristic we extract `owl:sameAs` links from DBpedia to BTC2014 so we have a complete set of linked instances between these datasets.

[4] http://km.aifb.kit.edu/projects/btc-2014/

Linked Data as Tables: The table generation approach is based on DBpediaAsTable[5] with proper modifications to fit BTC2014 triples. We create one table for each `rdf:type` and place instances of that type in rows, while the columns contain information about their properties. In practice, large tables are separated into several small tables by the limitation of 500 rows while columns are filtered by the density limitation which should be greater than 20%.

Property Matching: We argue that "`owl:sameAs` instances share similar values on similar properties". Once we obtain the `owl:sameAs` instances and similar values, similar properties could be inferred. Similar values are detected by computing similarity measures on literal, numeric and date cells. Afterwards, we can infer similar properties.

Parametrization: The final property correspondences are selected from a candidate set that is obtained from the property matching in last step. The selection is made by filtering property pairs using support threshold *su* and confidence threshold *co*. Property pair (p_1,p_2) holds with support *su* if *su*% of the `owl:sameAs` instances involved with p_1 or p_2 contain both p_1 and p_2. Property pair (p_1,p_2) holds with confidence *co* if *co*% of value pairs on (p_1,p_2) share similar values. We divide our gold standard into a learning set and a testing set. A genetic learning algorithm is applied on the learning set to obtain the proper values for *su* and *co*.

Result. We use three string-based metrics, Jaccard, Levenshtein and ExactEqual as baselines to compare with our approach. All metrics are applied on the testing set to find equivalent properties between BTC2014 and DBpedia. The results and the comparison is shown in Table 1.

Experiments	True Positive	False Positive	GS	Pre	Rec	F1
Instance-based property matching	84	23	85	0.785	0.988	0.875
Levenshtein	52	52	85	0.5	0.612	0.550
Jaccard	52	91	85	0.364	0.612	0.456
ExactEqual	32	0	85	1.0	0.376	0.547

Table 1. The results on property matching between BTC2014 and DBpedia.

The proposed approach can effectively match the property pairs which share similar values such as "landArea" with "areaTotal" and "diedIn" with "deathPlace". However, similar values also lead to wrong matchings such as "happenedOnDate" with "date", "capital" with "largestCity" and "hasPhotoCollection" with "label" which require more semantic matching on property labels than on their values.

References

[1] M. Cheatham and P. Hitzler. The properties of property alignment. In *Proc. of the 9th Int.'l Workshop on Ontology Matching (OM)*, pages 13–24, 2014.

[2] M. Nentwig, M. Hartung, A.-C. N. Ngomo, and E. Rahm. A Survey of Current Link Discovery Frameworks. *Semantic Web Journal*, 2015.

[5] http://wiki.dbpedia.org/services-resources/downloads/dbpedia-tables

RinsMatch: a suggestion-based instance matching system in RDF Graphs

Mehmet Aydar and Austin Melton

Kent State University, Department of Computer Science, USA
{maydar,amelton}@kent.edu

Introduction. In this paper, we present RinsMatch (RDF Instance Match), a suggestion-based instance matching tool for RDF graphs. RinsMatch utilizes a graph node similarity algorithm and returns to the user the subject node pairs that have similarities higher than a defined threshold. If the user approves the matching of a node pair, the nodes are merged. Then more instance matching candidate pairs are generated and presented to the user based on the common predicates and neighbors of the already matched nodes. RinsMatch then reruns the similarity algorithm with the merged RDF node pairs. This process continues until there is no more feedback from the user and the similarity algorithm suggests no new matching candidate pairs.

In our previous study [1], we proposed an algorithm for computation of entity similarities of an RDF graph using graph locality, neighborhood similarity, and the Jaccard measure. In the current study we use the proposed RDF entities similarity algorithm for pairing entities which may be merged if approved by the user. We make a similar assumption like the similarity flooding (SF) algorithm proposed in [2], that elements of two graphs are similar when their adjacent elements are similar. Comparing to SF, our technique requires more user interactions and more iterations for computation of entity similarity, but each time the similarity algorithm runs, it produces more accurate results assuming the user provided accurate feedback. Also, merging the RDF nodes reduces the size of the input data graph that the algorithm operates on, yielding less complexity each time.

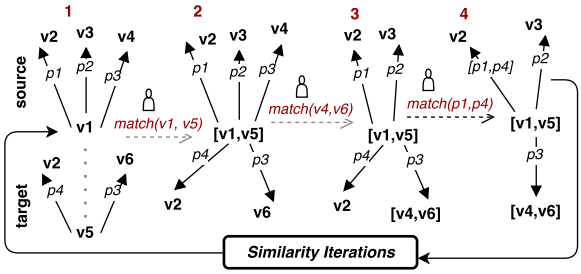


Fig. 1. Instance matching process

User Interaction. RinsMatch presents the subject node pairs that have similarities higher than a defined threshold to the user for possible instance matching. The threshold is a configurable parameter and may be determined by the user. If $s1$ and $s2$ are two subject nodes which have similarity higher than the threshold, then we denote this pair by ($s1$,$s2$). If the user approves the matching of the subject node pair ($s1$,$s2$), then RinsMatch merges the two subjects into a single

subject node which we denote by [$s1$,$s2$], and then all the predicates from both merged subjects are retained by the newly created subject [$s1$,$s2$]. RinsMatch then checks the common neighbors and predicates of $s1$ and $s2$ and generates more instance matching candidate pairs by pairing the predicates $p1$ and $p2$ to get ($p1$,$p2$) if $p1$ and $p2$ are connected to a common object from both $s1$ and $s2$. It also pairs the object nodes ($o1$,$o2$) which are connected with a common predicate by $s1$ and $s2$. RinsMatch then presents the generated matching candidate pairs to the user and merges the pairs to get [$p1$,$p2$] and [$o1$,$o2$] if the user approves that they match. RinsMatch then reruns the RDF node similarity algorithm on the new RDF graph formed by merging matching entities. The steps above are repeated until there is no more feedback from the user and no new matching pairs suggested by the matching algorithm. Figure 1 shows an example of the instance matching and merging process. As shown in the figure, based on the similarities found from the similarity iterations, at phase 1 RinsMatch suggests matching the subject nodes ($v1$,$v5$), and they are merged to get [$v1$,$v5$] with the approval of the user. On phase 2, the algorithm checks the common predicates of the new node [$v1$,$v5$]. Seeing that it connects to the neighbor nodes $v4$ and $v6$ with the common predicate $p3$, RinsMatch merges the nodes $v4$ and $v6$ to get [$v4$,$v6$] once the user approves. On phase 3, the common neighbors of the new node [$v1$,$v5$] are checked. Seeing that [$v1$,$v5$] is connected to a common neighbor $v2$ with the predicates $p1$ and $p4$, then RinsMatch presents the pair ($p1$,$p4$) to the user, and they are merged upon approval by the user to get [$p1$,$p4$]. The output graph of phase 3 is input to phase 1, and the similarity iterations are repeated until the optimum similarities and instance matching pairs are found.

Evaluation. We conducted preliminary experiments based on a subset of DBpedia and a subset of SemanticDB, a Semantic Web content repository for Clinical Research and Quality Reporting. For verification, we duplicated the original dataset and changed the names of the nodes in the duplicated dataset by following a specific naming pattern. We used the original dataset as the source, and the duplicated dataset as the target for the instance matching process, and we leveraged the node naming pattern for verification. To summarize our experiments: for the DBpedia, the source dataset had 90 triples with 60 distinct subject and predicate nodes. 100% of the nodes were matched to a target graph node semi-automatically. The algorithm generated 20 instance matching candidates with 85% accuracy. For SemanticDB, the source dataset had 2500 triples with 520 distinct subject and predicate nodes. 86% of the nodes were matched to a target graph node semi-automatically. The algorithm generated 310 instance matching candidates with 95% accuracy.

References

1. Mehmet Aydar, Serkan Ayvaz, and Austin C Melton. Automatic weight generation and class predicate stability in rdf summary graphs. In *Workshop on Intelligent Exploration of Semantic Data (IESD2015), co-located with ISWC2015*, 2015.
2. Sergey Melnik, Hector Garcia-Molina, and Erhard Rahm. Similarity flooding: A versatile graph matching algorithm and its application to schema matching. In *Data Engineering, 2002. Proceedings. 18th International Conference on*, pages 117–128. IEEE, 2002.

Triple-based Similarity Propagation for Linked Data Matching

Eun-kyung Kim, Sangha Nam, Jongseong Woo, Sejin Nam, and Key-Sun Choi

School of Computing, KAIST, Republic of Korea
{kekeeo,namsangha,woo88,namsejin,kschoi}@world.kaist.ac.kr

Abstract. In this paper, we propose an approach for mapping properties in two RDF datasets between different languages, using a triple-based similarity propagation that can be adapted to find potential property matches. This approach does not need any language dependent information during the process, and thus can be applied to arbitrary languages without requiring translation.

1 Introduction

Linked Data aims to extend the Web by publishing various open datasets as RDF and establishing connections between them. DBpedia exploits the huge amount of information contained in Wikipedia and creates a comprehensive dataset by integrating information from many different Wikipedia editions according to an ontology maintained by the community. Due to the interdisciplinary nature and the enormous breadth of coverage of Wikipedia, DBpedia is regarded as one of the central interlinking-hubs of Linked Data [1]. In this paper, we propose an approach for mapping *properties* across DBpedia RDF datasets written in the two languages using a triple-based similarity propagation that can be adapted to find potential property matches without any translation task.

2 Proposed Approach

The proposed approach has two steps: 1) findings the equivalent *subject* and *object* values across datasets at the entity-level, which is represented in the form triples, that are connected by `owl:sameAs` links, and then considering the associated *properties* to have the potential to be equivalent. 2) Then, using a small number of identified matches as seeds to exploit the conceptual-level alignments to identify and estimate semantic relatedness of *properties*. Often, the conceptualizations of triples (from instance triples) are efficient in terms of coverage of alignment, but their result may be dependent on recognizing entities and their type. The types of an entity may not always be present in the dataset. The 'similarity flooding approach' [2] propagates the similarities between concepts to refine the matching results. For example, two apparently different entities from two ontologies are similar when their neighboring concepts are similar.

Experiments: The goal of this experiment is to align language-local properties (i.e., DBpedia Korean property in this case) with the ontological properties of DBpedia in English. Three human annotators aligned 1,000 DBKP to DBOP, if the meaning of two properties was similar. We used the majority vote to determine the correct mapping results.

Table 1. *cf* is the confidence score of the derived property pairs. **I**, **P1**, and **P0** represent the three kinds of propagation scale strategies. **I** denotes cases in which the alignment process is done without using propagation technique. **P1** denotes results obtained from the similarity propagation with the seed with a cf=1, whereas cases with a **P0** executes the propagation step with a larger seed with a cf >=0. **#(M)** signifies the number of newly discovered matches, and **P**, **R**, and **F** means precisions, recalls and F1-scores, respectively.

	(**I**) w/o prop				(**P1**) prop:seed.θ=1				(**P0**) prop:seed.θ >=0			
cf.	#(**M**)	P	R	F	#(**M**)	P	R	F	#(**M**)	P	R	F
1	13	100	0.96	1.91	25	100	0.96	1.91	23	95.65	1.3	2.57
0.9	42	95.24	2.97	5.76	50	94.44	2.52	4.91	56	94.64	3.14	6.07
0.7	98	96.94	7.05	13.14	118	93.26	6.16	11.55	121	97.52	6.99	13.04
0.5	151	96.69	10.83	19.48	226	90.70	11.57	20.53	199	97.99	11.55	20.66
0.4	188	95.74	13.35	23.44	282	91.12	14.47	24.97	246	95.93	13.97	24.39
0.3	222	95.5	15.73	27.01	386	88.66	19.14	31.48	306	94.12	17.05	28.87
0.2	269	94.42	18.84	31.42	538	82.80	22.85	35.81	381	89.5	20.19	32.95
0.1	322	92.55	22.11	35.69	863	76.01	27.97	40.89	505	84.95	25.4	39.11
0	668	75.15	37.24	49.80	3,166	59.35	47.11	52.52	896	74.33	39.43	51.53

Analysis: The preliminary experiment between the English and the Korean DBpedia has shown that the propagated connectives improve the recall and F1-score measures required to find mapping pairs of properties by taking into account instance types in order to discover new mapping candidates. We see this as the initial step towards enhancing multilingualism in Linked Open Data.

Acknowledgement This work was supported by Institute for Information & communications Technology Promotion(IITP) grant funded by the Korea government(MSIP) (No. R0101-15-0054, WiseKB: Big data based self-evolving knowledge base and reasoning platform)

References

1. Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P.N., Hellmann, S., Morsey, M., van Kleef, P., Auer, S., Bizer, C.: DBpedia - a large-scale, multilingual knowledge base extracted from wikipedia. Semantic Web Journal (2014)
2. Melnik, S., Garcia-Molina, H., Rahm, E.: Similarity flooding: A versatile graph matching algorithm and its application to schema matching. In: Agrawal, R., Dittrich, K.R. (eds.) ICDE. pp. 117–128. IEEE Computer Society (2002), `http://dblp.uni-trier.de/db/conf/icde/icde2002.html#MelnikGR02`

An Effective Configuration Learning Algorithm for Entity Resolution

Khai Nguyen and Ryutaro Ichise

The Graduate University for Advanced Studies, Japan
National Institute of Informatics, Japan
{nhkhai,ichise}@nii.ac.jp

1 Introduction

Entity resolution is the problem of finding co-referent instances, which at the same time describe the same topic. It is an important component of data integration systems and is indispensable in linked data publication process. Entity resolution has been a subject of extensive research; however, seeking for a perfect resolution algorithm remains a work in progress.

Many approaches have been proposed for entity resolution. Among them, supervised entity resolution has been revealed as the most accurate approach [6, 2]. Meanwhile, configuration-based matching [2, 3, 5, 4] attracts most studies because of its advantages in scalability and interpretation.

In order to match two instances of different repositories, configuration-based matching algorithms estimate the similarities between the values of the same attributes. After that, these similarities are aggregated into one matching score. This score is used to determine whether two instances are co-referent or not. The declarations of equivalent attributes, similarity measures, similarity aggregation, and acceptance threshold are specified by a matching configuration, which can be automatically optimized by a learning algorithm. Configuration learning using genetic algorithm has been a research topic of some studies [2, 5, 3]. The limitation of genetic algorithm is that it costs numerous iterations for reaching the convergence. We propose *cLearn* as a heuristic algorithm that is effective and more efficient. *cLearn* can be used to enhance the performance of any configuration-based entity resolution system.

2 Approach

A configuration specifies the property mappings, similarity measures, similarity aggregation strategy, and matching acceptance threshold. Property mappings and similarity measures are combined together into similarity functions. Given series of initial similarity functions, similarity aggregation options, and the labeled instances pairs, the mission of *cLearn* is to select the optimal configuration.

cLearn begins with the consideration of each single similarity function and then checks their combinations. When checking the new combination this algorithm applies a heuristic for selecting most potentially optimal configuration. Concretely, the heuristic accepts the new combination if only its performance

Table 1. F1 score of the compared systems on OAEI 2010.

Training size	System	Sider-Drugbank	Sider-Diseasome	Sider-DailyMed	Sider-DBpedia	DailyMed-DBpedia
5%	ScSLINT+*cLearn*	**0.911**	**0.824**	**0.777**	0.6414	**0.424**
	Adaboost	0.903	0.794	0.733	0.641	0.375
Varied by subset	ScSLINT+*cLearn*	**0.894**	**0.829**	**0.722**		
	ObjectCoref	0.464	0.743	0.708		

is better than that of the combined elements. This heuristic is reasonable as a series of similarity functions that reduces the performance has little possibility of generating a further combination with improvement. In addition to finding for similarity functions, the algorithm also optimizes the similarity aggregator and matching acceptance threshold.

cLearn is implemented as part of ScSLINT framework, and its source code is available at http://ri-www.nii.ac.jp/ScSLINT.

3 Evaluation

Table 1 reports the comparison between *cLearn* and other supervised systems, including ObjectCoref [1] and Adaboost-based instance matching system [6]. OAEI 2010 dataset is used and the same amount of training data is given to each pair of compared systems. According to this table, *cLearn* consistently outperforms other algorithms.

cLearn is efficient as the average numbers of configurations that *cLearn* has to check before stopping is only 246. This number is promising because it is much smaller than that of using genetic algorithm, which is reported in [2] with a recommendation of 500 configurations for each iteration.

With the effectiveness, potential efficiency, and small training data requirement of *cLearn* on a real dataset like OAEI 2010, we believe that *cLearn* has promising application in supervised entity resolution, including using active learning strategy to even reduce the annotation effort.

References

[1] Hu, W., Chen, J., Cheng, G., Qu, Y.: Objectcoref & falcon-ao: results for oaei 2010. In: 5th Ontology Matching. pp. 158–165 (2010)

[2] Isele, R., Bizer, C.: Active learning of expressive linkage rules using genetic programming. Web Semantics: Science, Services and Agents on the World Wide Web 23, 2–15 (2013)

[3] Ngomo, A.C.N., Lyko, K.: Unsupervised learning of link specifications: Deterministic vs. non-deterministic. In: 8th Ontology Matching. pp. 25–36 (2013)

[4] Nguyen, K., Ichise, R., Le, B.: Interlinking linked data sources using a domain-independent system. In: 2nd JIST. LNCS, vol. 7774, pp. 113–128. Springer (2013)

[5] Nikolov, A., d'Aquin, M., Motta, E.: Unsupervised learning of link discovery configuration. In: 9th ESWC. LNCS, vol. 7295, pp. 119–133. Springer (2012)

[6] Rong, S., Niu, X., Xiang, W.E., Wang, H., Yang, Q., Yu, Y.: A machine learning approach for entity resolution based on similarity metrics. In: 11th ISWC. LNCS, vol. 7649, pp. 460–475. Springer (2012)

Search Space Reduction for Post-Matching Correspondence Provisioning

Thomas Kowark and Hasso Plattner

Hasso Plattner Institute, Potsdam, Germany
{firstname.lastname}@hpi.de,
http://www.hpi.de

If users participate in ontology matching, the goal always is to minimize the amount of necessary interactions while maximizing the gains in alignment quality [2]. Interaction can either happen pre-matching (selection of matching systems or parameter tuning), during the matching process (judging intermediate results or providing sample correspondences), or post-matching (detecting incorrect correspondences and providing missing ones). In this paper, we evaluate an approach that aims to reduce post matching interactions by exploiting concept proximity within ontologies. An initial analysis of reference alignments available for OAEI revealed that, if a correspondence for one element (class or property) of an ontology exists, the probability that a correspondence also exists for a closely connected element is higher than for unconnected elements. Based on this finding, we extracted the closeness criteria depicted in Figure 1. For evaluation, we applied the criteria on candidate alignments that were created by top-performing systems of OAEI 2014 for the anatomy, library, and conference tracks. For each criterion, we determined which elements it would add to the task set, i.e., the selection of ontology elements a user should provide correspondences for. Based on these task sets (UT) we calculated the expected number of interactions (IE) it would on average take to provide all included correspondences (IC), if elements were presented to the user at random. To assess whether our selection technique is viable, we further compared this value to the amount of interactions it would take users on average to provide the same amount of missing correspondences, if tasks were randomly selected from the entirety of elements that are not included in correspondences after initial, automatic matching.

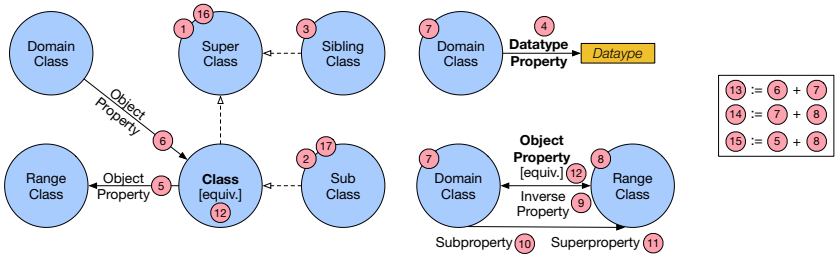


Figure 1. Connections considered for element proximity. Matched elements are depicted bold, ignored entities in italic. Visual Notation for OWL Ontologies (VOWL)[1]

The ratio between the two values is called task set compression. Minimal criteria sets denote the closeness criteria, which yield the corresponding task sets. Since we strive for minimization of user tasks, only the smaller ontology in terms of concept count was considered. As shown in Table 1, an average task set reduction of 60% could be achieved for the conference ontologies of OAEI, while increasing the recall from 0,62 to 0,956. For taxonomy-like ontologies, such as the ones used in the library and anatomy tracks, only marginal compression or even increase in interaction expectancy was achieved. Future work will therefore focus on such cases by finding other, more suitable task selection criteria and adapting existing ones, e.g., by limiting the depth of hierarchy traversal for class relationships. Furthermore, correspondences generated through different matcher settings (high precision vs. high recall) could be explored in addition to criteria based solely on ontology structures in order to yield smaller task sets with an increased potential success ratio for user interactions.

ontologies	$\|UT\|$	IC	IE	Compression	Minimal Criteria Sets	R_{cand}	R_{comp}
cmt-conference	12	4	10	0,2	[9, 17]	0,6	0,87
confof-conference	11	3	9	0,19	[13]	0,73	0,93
conference-edas	36	5	31	0,39	[1, 2]	0,65	0,94
ekaw-conference	38	8	35	0,51	[3, 6, 16]	0,6	0,92
conference-iasted	51	7	46	0,57	[1, 2, 5]	0,36	0,86
sigkdd-conference	13	4	11	0,25	[4, 9, 12]	0,6	0,87
confof-cmt	31	9	29	0,48	[2, 4, 6, {7, 8, 14}]	0,38	1
cmt-edas	27	4	22	0,35	[6]	0,69	1
cmt-ekaw	38	5	33	0,49	[5, 9, 17]	0,55	1
cmt-iasted	36	0	36	0,44	[]	1	1
sigkdd-cmt	7	1	4	0,13	[2]	0,92	1
confof-edas	26	8	24	0,43	[1, 3, 5, 9]	0,58	1
confof-ekaw	18	4	15	0,33	[1, 13, 15]	0,8	1
confof-iasted	30	5	26	0,45	[1, 3, 15]	0,44	1
confof-sigkdd	16	4	13	0,25	[4, 17]	0,57	1
ekaw-edas	63	11	59	0,71	[9, 13, 17]	0,52	1
edas-iasted	35	8	32	0,28	[2]	0,53	0,95
sigkdd-edas	20	5	18	0,37	[2, 4]	0,6	0,93
ekaw-iasted	73	3	56	0,76	[2, 15], [2, 13]	0,7	1
sigkdd-ekaw	22	4	18	0,32	[3, 15]	0,64	1
sigkdd-iasted	36	0	36	0,58	[]	0,87	0,87
avg(conference)	30,4	4,8	26,8	0,404		0,635	0,956
mouse-human	683	57	672	1,09	[2, 3, 16]	0,9	0,94
stw-thesoz	3604	169	3584	0,97	[2, 3]	0,78	0,84
fma-nci	1011	216	1007	1,08	[2, 3, 16]	0,85	0,93
fma-snomed	3485	1997	3484	0,99	[2, 3]	0,71	0,95
nci-snomed	10008	2281	10005	0,94	[1, 2, 3, 12]	0,67	0,82

Table 1. Overview about the maximal task reduction that could be achieved using minimal criteria sets. The used criteria are numbered according to Figure 1. R_{cand} is the recall achieved by the automatic matcher, R_{comp} the recall after user interaction.

References

1. Lohmann, S., Negru, S., Haag, F., Ertl, T.: VOWL 2: User-Oriented Visualization of Ontologies. In: Proceedings of the 19th International Conference on Knowledge Engineering and Knowledge Management. pp. 266–281. EKAW '14 (2014)
2. Shvaiko, P., Euzenat, J.: Ontology matching: State of the art and future challenges. IEEE Trans. on Knowl. and Data Eng. 25(1), 158–176 (Jan 2013)

Automatic mapping of Wikipedia categories into OpenCyc types*

Aleksander Smywiński-Pohl[1,2] and Krzysztof Wróbel[1,2]

[1] Jagiellonian University, Faculty of Management and Social Communication
[2] AGH University of Science and Technology, Faculty of Computer Science, Electronics and Telecommunications

Abstract. The aim of the research presented in the article is the mapping between the English Wikipedia categories and OpenCyc types. The mapping algorithm is heuristic and it takes into account structural similarities between the categories and the corresponding types. The achieved mapping precision ranges from 82 to 92 % (depending on the evaluation scheme), recall from 67 to 76%. The results of the algorithm and its code are available at `http://cycloped.io`.

1 Approach

The aim of this research is automatic mapping of Wikipedia categories into OpenCyc [1] types. Although Wikipedia category system is hierarchical in nature, it is more like a thesaurus than a classification scheme [5], since it lacks any clear-defined hierarchical structure [4]. By mapping the categories into OpenCyc types we will be able to levarage the well defined structure of that ontology in Wikipedia-related information extraction tasks.

The automatic mapping of categories is divided into three stages. In the first stage the categories are pre-processed, in order to filter-out the uninteresting categories. In the second stage for each category a set of candidate mappings is generated and in the last stage disambiguation is performed by comparing the context of the category with the contexts of the candidate types. As such it is similar to the method employed in YAGO for mapping the categories into WordNet synsets [3].

The disambiguation is based on structural similarities between the OpenCyc ontology and Wikipedia category system treated as a taxonomy. The primary means for structuring Wikipedia is the *inclusion* relation that holds between categories and articles as well as categories themselves. In the first case, if the article represents an entity, the inclusion in a category might be approximated by *instantiation* relation, while in the second case the inclusion of category might be approximated by *specialization* relation. *Instantiation* and *specialization* are strictly defined in OpenCyc and are the primary means for structuring its contents. Checking if inclusion of articles and categories in the category that is being

* This work was supported by the Faculty of Management and Social Communication, Jagiellonian University in Krakow.

mapped has a corresponding instantiation and specialization assertions stated in OpenCyc provides evidence for validity of a given candidate mapping.

2 Results

Out of 616 thousand of categories with plural noun-heads we were able to assign some corresponding type to 484 thousand categories (78.6%). We have manually validated 600 mappings in order to assess the quality of the category mapping algorithm. We assumed that there is up to one valid OpenCyc type for each Wikipedia category. We have not assigned any type if the category was ambiguous or should be filtered out as administrative. In cases the algorithm assigned some types to such categories, they were treated as false positives. For the other categories we have either accepted the mapping provided by the algorithm or manually assigned the correct mapping in cases when the algorithm's decision was invalid.

We measured the performance of the algorithm using standard information retrieval measures of precision and recall, employing two evaluation scenarios. In the first one strict equivalence between the results obtained by the algorithm and the reference mapping was required and in the second, we have extended the set of true positives, by including results that were either specializations or generalizations of the terms defined in the reference set. In the first scenario we have obtained 82.5% precision, 67.5% recall and 74.2% F_1 and in the second we have obtained 92.9% precision, 76.1% recall and 83.6% F_1.

The results of the algorithm and the source code are available at `http://cyclo-ped.io`. We plan to extend the mapping and classification into other natural languages, as well as automatically extend the OpenCyc taxonomy. Although the results of the automatic mapping are worse than manually established correspondence from our past efforts [2], the achieved coverage is much better. Moreover the algorithms allow for providing new mappings when Wikipedia grows, making it very useful for converting it into computable knowledge base.

References

1. Lenat, D.B.: CYC: A large-scale investment in knowledge infrastructure. Communications of the ACM 38(11), 33–38 (1995)
2. Pohl, A.: Classifying the Wikipedia Articles into the OpenCyc Taxonomy. In: Rizzo, G., Mendes, P., Charton, E., Hellmann, S., Kalyanpur, A. (eds.) Proceedings of the Web of Linked Entities Workshop in conjuction with the 11th International Semantic Web Conference. pp. 5–16 (2012)
3. Suchanek, F., Kasneci, G., Weikum, G.: YAGO: a core of semantic knowledge. In: Williamson, C., Zurko, M.E., Patel-Schneider, P., Shenoy, P. (eds.) Proceedings of the 16th international conference on World Wide Web. pp. 697–706. ACM (2007)
4. Suchecki, K., Salah, A.A.A., Gao, C., Scharnhorst, A.: Evolution of Wikipedia's Category Structure. Advances in Complex Systems 15(supp01) (2012)
5. Voss, J.: Collaborative thesaurus tagging the Wikipedia way. arXiv preprint cs/0604036 (2006)

Exploiting Multilinguality For Ontology Matching Purposes

Mauro Dragoni

FBK–IRST, Trento, Italy
dragoni@fbk.eu

1 Introduction

The alignment between linguistic artifacts like vocabularies, thesauri, etc., is a task that has attracted considerable attention in recent years [1][2]. With very few exceptions, however, research in this field has primarily focused on the development of monolingual matching algorithms. As more and more artifacts, especially in the Linked Open Data realm, become available in a multilingual fashion, novel matching algorithms are required.

Indeed, in the case of a multilingual environment, there are some peculiarities that can be exploited in order to relax the classic schema matching task:

- the use of multilinguality permits to reduce the problems raised when two different concepts have the same label; indeed, the probability for two diverse concepts to have the same label across several languages is very low;
- multilingual artifacts provide term translations that have already been adapted to the represented domains; therefore, the human creators of a multilingual artifact put a lot of their cultural heritage in choosing the right terms for the each concept.

In this paper, we present a work exploiting the two aspects described above in order to build a multilingual ontology approach for defining mappings between multilingual ontologies. Such an approach, extending the one presented in [3], has been evaluated on domain-specific use cases belonging to the agriculture and medical domains.

2 An Approach for the Matching of Multilingual Thesauri

The proposed approach is based on the exploitation of the labels associated with each concept defined in an ontology. Let us consider two ontologies: (i) a source ontology containing the elements that have to be mapped, and a target ontology used as reference for creating the mappings. The proposed approach has been built by taking inspiration from IR techniques and it exploits the creation of indexes for identifying candidate mappings.

The process is split in two different phases: (i) in the first one, we created the index containing information about the target ontology represented in a structured way; while, (ii) in the second phase, we build queries using information contained in the source ontology for retrieving a rank representing the candidate mappings that we may define between the two thesauri.

Firstly, we extract the whole set of labels from the target ontology and, after a set of preprocessing activities, each concept "C" of the target ontology is transformed into a structured representation containing all multilingual labels describing "C", and all multilingual labels describing concepts belonging to the context of "C" that is the set of concepts directly connected with "C". Such labels are then stored into an index. Then, in the second phase, from each entity of the source index the set of its labels is extracted. A query containing such labels is composed and performed on the index built during the first phase. A rank containing n suggestions ordered by their confidence score is returned by the system and it is used as input for the creation of the mapping that may be done manually from domain experts or automatically by the system.

3 Concluding Remarks

The approach has been evaluated on a set of six multilingual ontologies, coming from the agricultural and medical domains, for which gold standards containing the mappings were available. Then, it has been compared with the previous one presented in [3].

Mapping Set	**# of Mappings**	**Prec. v1**	**Rec. v1**	**F-Measure v1**	**Prec. v2**	**Rec. v2**	**F-Measure v2**
Eurovoc → Agrovoc	1297	0.816	0.874	0.844	0.897	1.000	0.946
Agrovoc → Eurovoc	1297	0.906	0.695	0.787	0.930	0.999	0.963
	Avg.	*0.861*	*0.785*	*0.821*	*0.914*	*1.000*	***0.955***
Gemet → Agrovoc	1179	0.909	0.546	0.682	0.850	0.999	0.918
Agrovoc → Gemet	1179	0.943	0.740	0.829	0.893	0.997	0.942
	Avg.	*0.926*	*0.643*	*0.759*	*0.872*	*0.998*	***0.931***
MDR → MeSH	6061	0.776	0.807	0.791	0.903	0.912	0.907
MeSH → MDR	6061	0.716	0.789	0.751	0.843	0.888	0.865
	Avg.	*0.746*	*0.798*	*0.771*	*0.873*	*0.900*	***0.886***
MDR → SNOMED	19971	0.621	0.559	0.588	0.739	0.826	0.780
SNOMED → MDR	19971	0.556	0.519	0.537	0.871	0.459	0.601
	Avg.	*0.589*	*0.539*	*0.563*	*0.805*	*0.643*	***0.715***
MeSH → SNOMED	26634	0.690	0.660	0.675	0.741	0.814	0.776
SNOMED → MeSH	26634	0.657	0.564	0.607	0.831	0.544	0.658
	Avg.	*0.674*	*0.612*	*0.642*	*0.786*	*0.679*	***0.729***

Table 1: Comparison between the results obtained by the previous version of the system and the proposed one.

References

1. Euzenat, J., Shvaiko, P.: Ontology matching. Springer (2007)
2. Bellahsene, Z., Bonifati, A., Rahm, E., eds.: Schema Matching and Mapping. Springer (2011)
3. Dragoni, M.: Exploiting multilinguality for creating mappings between thesauri. In: Proceedings of the 30th Annual ACM Symposium on Applied Computing. SAC 2015, ACM (2015) 382–387

Ontology Matching Techniques for Enterprise Architecture Models

Marzieh Bakhshandeh[1], Catia Pesquita[2] and José Borbinha[2]

[1] INESC-ID - Information Systems Group, Lisbon, Portugal
[2] LaSIGE, Faculdade de Ciências, Universidade de Lisboa, Portugal
[3] Instituto Superior Técnico, Universidade de Lisboa, Portugal
marzieh.bakhshandeh@ist.utl.pt,
cpesquita@di.fc.ul.pt,jose.borbinha@ist.utl.pt

Abstract. Current Enterprise Architecture (EA) approaches tend to be generic, based on broad meta-models that cross-cut distinct architectural domains. Integrating these models is necessary to an effective EA process, in order to support, for example, benchmarking of business processes or assessing compliance to structured requirements. However, the integration of EA models faces challenges stemming from structural and semantic heterogeneities that could be addressed by ontology matching techniques. For that, we used AgreementMakerLight, an ontology matching system, to evaluate a set of state of the art matching approaches that could adequately address some of the heterogeneity issues. We assessed the matching of EA models based on the ArchiMate and BPMN languages, which made possible to conclude about not only the potential but also of the limitations of these techniques to properly explore the more complex semantics present in these models.

Enterprise Architecture (EA) is a practice to support the analysis, design and implementation of a business strategy in an organization, considering its relevant multiple domains. In recent years, a variety of Enterprise Architecture [5] languages have been established to manage the scale and complexity of this domain. Integration of EA models is necessary to support EA processes, however structural and semantic heterogeneities hinder integration. Ontology matching has been proposed as a useful technique to help address this challenge [4]. Ontologies and associated techniques are increasingly being recognized as valuable tools in the EA domain (e.g., [1]).

To evaluate the applicability of ontology matching techniques to address the heterogeneity between EA models, we have selected four case studies that demonstrate heterogeneity challenges at the model level. Cases 1 and 2 showcase *Abstraction Level Incompatibilities* between models encoded in different languages (ArchiMate and BPMN), that represent similar situations. Cases 3 and 4 illustrate both *Abstraction Level* and *Element Description* heterogeneties between models using the same language, where both pairs of models represent the same situation encoded by different modelers.

To support the matching tasks we have used AgreementMakerLight (AML) [2],

an ontology matching system that is extensible and implements several state of the art ontology matching algorithms. We extended AML to produce subclass mappings. The generated alignments were manually evaluated.

The four case studies and their matching using a combination of ontology matching algorithms illustrate the challenges and opportunities in their application to addressing EA heterogeneities. As expected, string and word based techniques are effective at capturing the mappings between equivalent individuals who share similar names. However, when equivalent individuals had dissimilar labels, for which WordNet extension did not produce any shared synonyms, the applied algorithms failed. Regarding *Abstraction Level Incompatibilities*, the results were related to the complexity of the models. In simpler model matching tasks, the Subclass Matcher approach had a good performance, identifying 75% of the subclass mappings. However, in more complex tasks performance is reduced. Since the evaluated approaches relied only on model information to perform matching, there was no practical difference between matching models using the same or different languages.

We consider that the main limitation of the employed matching techniques was their inability to explore a considerable portion of the information modelled in the ontologies. In order to extend the application of ontology matching techniques to the EA domain, ontology matching systems need to be able to explore this semantic richness by producing semantic matching approaches that go beyond current strategies which are mostly WordNet based [3]. In recent years, ontology matching systems have had a growing interest in terms of reasoning capabilities, and we propose that a combination of these strategies with pattern-based complex matching approaches [6] may provide improved solutions to the EA model integration challenge.

References

1. Antunes, G., Bakhshandeh, M., Mayer, R., Borbinha, J., Caetano, A.: Using ontologies for enterprise architecture integration and analysis. Complex Systems Informatics and Modeling Quarterly (1), 1–23 (2014)
2. Faria, D., Pesquita, C., Santos, E., Cruz, I.F., Couto, F.M.: Agreementmakerlight results for OAEI 2013. In: OM. pp. 101–108 (2013)
3. Giunchiglia, F., Autayeu, A., Pane, J.: S-match: an open source framework for matching lightweight ontologies (2010)
4. Karagiannis, D., Höfferer, P.: Metamodeling as an integration concept. In: Software and Data Technologies, pp. 37–50. Springer (2008)
5. Lankhorst, M., et al.: Enterprise Architecture at Work: Modelling, Communication and Analysis (The Enterprise Engineering Series). Springer (2013)
6. Ritze, D., Meilicke, C., Sváb-Zamazal, O., Stuckenschmidt, H.: A pattern-based ontology matching approach for detecting complex correspondences. In: ISWC Workshop on Ontology Matching, Chantilly (VA US). pp. 25–36. Citeseer (2009)

[1] **Acknowledgements:**This work was supported by national funds through Fundação para a Ciência e a Tecnologia (FCT) with references UID/CEC/50021/2013 and UID/CEC/00408/2013.

MOSEW: A Tool Suite for Service Enabled Workflow

Mostafijur Rahman and Wendy MacCaull

Department of Mathematics, Statistics and Computer Science,
St. Francis Xavier University, Antigonish, NS, Canada
{x2013ici,wmaccaul}@stfx.ca

Abstract. Recently our research group introduced the notion of Service Enabled Workflow (SEW) with the integration of Semantic Web Service (SWS) and Workflow. In this paper, we present a Service Oriented Architecture (SOA)-based integrated software tool suite called MOSEW that provides functionalities to design and develop ontology based Quality of Service (QoS) aware SEWs.

1 Introduction

Service Enabled Workflow (SEW) [1] is relatively a new concept in the area of Semantic Web-based research. SEW considers workflow as a collection of tasks with specific control flow where tasks are carried out as services. While it has a lot of potential, SEW still requires a great deal of maturity and support of tools to become an industry standard. The MOSEW tool suite provides functionalities to design and develop ontology based SEWs where QoS-aware SWSs are discovered, selected and executed dynamically by a mobile agent for some of the tasks in a workflow to complete the overall execution.

2 QoS aware SWS Discovery, Selection and Execution

To manage the QoS specifications of the services effectively and utilize them to improve the service discovery approach, we designed a QoS conceptual model and integrated it into the OWL-S 1.2 framework. To read the OWL-S 1.2 service descriptions and execute the WSDL [2] service grounding, motivated by the efforts [2] and [4], we designed and developed the OWL API based OWL-S API that provides a Java API. The ontology based core matching algorithm, which extends algorithm [3] consists of two parts: basic functional (I/O) property-based matching and non-functional property (QoS)-based matching. We developed a ranking algorithm and placed it on top of the Service Discovery Engine that executes the semantic matchmaking algorithm. To access and execute the service, we used the grounding information of an OWL-S service.

3 MOSEW Architecture

We used this QoS-aware web service discovery, selection and execution approach in the Service Discovery and Execution engines of the MOSEW tool suite. The tool suite allows consumers to graphically define specifications of workflow tasks using ontology guided user interfaces and execute the workflows dynamically by a smart phone based software agent. Fig 1 shows the MOSEW architecture.

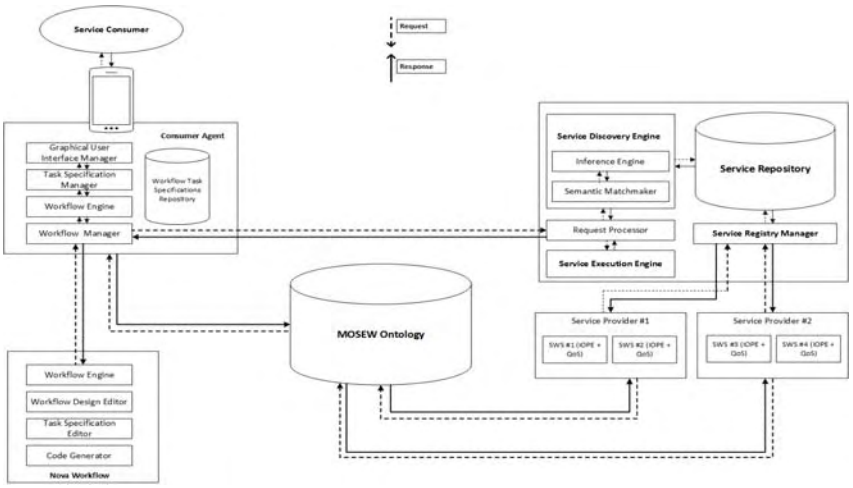


Fig. 1. MOSEW Architecture

4 Conclusion and Future Work

In this paper, we present MOSEW, a SOA-based integrated software tool suite that is used to design and develop SEWs running on mobile devices. We achieved this functionality through SWS discovery, selection, execution and semi-automatic run time composition. This type of service composition is time consuming and less flexible. The automatic service composition method generates the process model automatically or locates the correct services if an abstract process model is presented. In future, we will extend the MOSEW tool suite to support automatic service composition.

References

1. Altaf Hussain and Wendy MacCaull, Context Aware Service Discovery and Service Enabled Workflow, pp. 45-48. Canadian Semantic Web Symposium, 2013.
2. Evren Sirin and Bijan Parisa, The OWL-S Java API. Alternate Paper Tracks, 2003.
3. Massimo Paolucci et al. Semantic Matching of Web Services Capabilities, pp. 333-347. International Semantic Web Conference, 2002.
4. OWL-S API, `http://projects.semwebcentral.org/projects/owl-s-api/`