Ontology Matching

OM-2014

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 the 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 will strive to improve academic awareness of industrial and final user needs, and therefore direct research towards those needs. Simultaneously, the workshop will serve to inform industry and user representatives about existing research efforts that may meet their requirements. The workshop will also investigate 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) 2014 campaign[2]. The particular focus of this year's OAEI campaign is on real-world specific matching tasks as well as on evaluation of interactive matchers and matchers for query answering. Therefore, the ontology matching evaluation initiative itself will provide 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 5 submissions for oral presentation and 9 submissions for poster presentation. 14 matching system participated in this year's OAEI campaign. Further information about the Ontology Matching workshop can be found at: `http://om2014.ontologymatching.org/`.

[1] `http://www.ontologymatching.org/`
[2] `http://oaei.ontologymatching.org/2014`

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
Ming Mao
Juanzi Li
Ernesto Jiménez-Ruiz
Axel Ngonga

October 2014

[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 & LIG, France
Ming Mao, Electronic Arts, USA
Ernesto Jiménez-Ruiz, University of Oxford, UK
Juanzi Li, Tsinghua University, China
Axel Ngonga, University of Leipzig, Germany

Program Committee

Alsayed Algergawy, Jena University, Germany
Michele Barbera, Spazio Dati, Italy
Zohra Bellahsene, LRIMM, France
Chris Bizer, University of Mannheim, Germany
Olivier Bodenreider, National Library of Medicine, USA
Michelle Cheatham, Write State University, USA
Marco Combetto, Informatica Trentina, Italy
Gianluca Correndo, University of Southampton, UK
Isabel Cruz, The University of Illinois at Chicago, USA
Jérôme David, INRIA & LIG, France
Stefan Dietze, L3S, Germany
Alfio Ferrara, University of Milan, Italy
Avigdor Gal, Technion, Israel
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
Yannis Kalfoglou, Ricoh Europe plc, UK
Anastasios Kementsietsidis, IBM, USA
Patrick Lambrix, Linköpings Universitet, Sweden
Nico Lavarini, Expert System, Italy
Tatiana Lesnikova, INRIA, France
Vincenzo Maltese, University of Trento, Italy
Fiona McNeill, University of Edinburgh, UK
Christian Meilicke, University of Mannheim, Germany
Andriy Nikolov, Open University, UK
Leo Obrst, The MITRE Corporation, USA
Heiko Paulheim, University of Mannheim, Germany
Yefei Peng, Google, USA
Andrea Perego, European Commission - Joint Research Centre, Italy

Catia Pesquita, University of Lisbon, Portugal
Alessandro Solimando, University of Genova, Italy
Umberto Straccia, ISTI-C.N.R., Italy
Ondřej Zamazal, Prague University of Economics, Czech Republic
Cássia Trojahn, IRIT, France
Giovanni Tummarello, Fondazione Bruno Kessler - IRST, Italy
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

PART 1 - Technical Papers

PART 2 - OAEI Papers

PART 3 - Posters

A Categorical Approach to Ontology Alignment

Mihai Codescu, Till Mossakowski, and Oliver Kutz

Institute of Knowledge and Language Engineering
Otto-von-Guericke University of Magdeburg, Germany

Abstract. Ontology matching and alignment is a key mechanism for linking the diverse datasets and ontologies arising in the Semantic Web. We show that category theory provides the powerful abstractions needed for a uniform treatment at various levels: semantics, language design, reasoning and tools. The Distributed Ontology Language DOL is extended in a natural way with constructs for networks of ontologies. We in particular show how the three semantics of Zimmermann and Euzenat can be uniformly and faithfully represented using these DOL language constructs. Finally, we summarise how the DOL alignment features are currently being implemented in the Ontohub/Hets ecosystem, including support for the OWL and Alignment APIs.

1 Introduction

Ontology matching and alignment is a key mechanism for linking the diverse datasets and ontologies arising in the Semantic Web. Matching based on statistical methods is a relatively developed field, with yearly competitions since 2004 comparing the various strengths and weaknesses of existing algorithms [20].

Ontology alignments express semantic correspondences between the entities of different ontologies. The correspondences of an alignment can be various relations, like equivalence, subsumption, disjointness or instance between entities of the ontologies, which can be named entities, like classes, roles, individuals, function symbols etc. or even complex concepts or terms.

The problem of giving an interpretation to alignments in terms of the semantics of the ontologies is complicated by the fact that the domains of interpretation of the two ontologies may be incompatible. Different ways of dealing with this problem exist in the literature. The first solution, called simple semantics in [23], is to assume that the domain of interpretation of the ontologies is uniform [4, 5]. The second solution, called *integrated semantics* in [23], is to assume the existence of a universal domain together with functions relating the domains of individual ontologies to the universal domain. This approach has been introduced in [21], under the name of integrated distributed description logics (IDDL). Finally, the domains of the individual ontologies can be related among themselves directly instead via a unique universal domain. This approach gives rise to the third semantics, called *contextualised semantics* in [23]. It was introduced in [23] as an attempt to generalise a number of existing semantic formalisms (distributed first-order logics (DFOL) [10], distributed description logics (DDL) [2] and contextualised ontologies (C-OWL) [3]) and later corrected to a relational semantics in [22]. Package-based description logics (PDL) [1] also fall in this semantic category. Moreover, [23] discusses the implications of these possible interpretations of alignments with respect to reasoning and composition of alignments.

A major problem with these approaches is their diversity. There exist some attempts for unification, which however remain unsatisfactory: there is no common syntax, no common semantic framework, and no common tool support. In this work, we show how category theory can provide such a unifying framework at various levels, improving previous related work [24, 15, 22, 11] which did not spell out details, and did not make the step from abstract description and case studies to language design and implementation.

2 General approach

The general representation and reasoning framework that we propose includes: 1) a declarative language to specify networks of ontologies and alignments, with independent control over specifying local ontologies and complex alignment relations, 2) the possibility to align heterogeneous ontologies, and 3) in principle, the possibility to combine different alignment paradigms (simple/integrated/contextualised) within one network.

Through category theory, we obtain a unifying framework at various levels:

semantic level We give a uniform semantics for distributed networks of aligned ontologies, using the powerful notion of *colimit*, while reflecting properly the semantic variation points indicated above.

(meta) language level We provide a uniform notation (based on the distributed ontology language DOL) for distributed networks of aligned ontologies, spanning the different possible semantic choices.

reasoning level Using the notion of colimit, we can provide reasoning methods for distributed networks of aligned ontologies, again across all semantic choices.[1]

tool level The tool `ontohub.org` provides an implementation of analysis and reasoning for distributed networks of aligned ontologies, again using the powerful abstractions provided by category theory.

logic level Our semantics is given for the ontology language OWL, but due to the abstraction power of the framework, it easily carries over to other logics used in ontology engineering, like RDFS, first-order logic or F-logic.

This shows that category theory is not only a powerful abstraction at the semantic level, but can properly guide language design and tool implementations and thus provide useful abstraction barriers from a software engineering point of view.

The distributed ontology language DOL is a metalanguage in the sense that it enables the reuse of existing ontologies as building blocks for new ontologies using a variety of structuring techniques, as well as the specification of relationships between ontologies. One important feature of DOL is the ability to combine ontologies that are written in different languages without changing their semantics. A formal specification of the language can be found in [17]. However note that syntax and semantics of DOL alignments is introduced in this paper for the first time.

[1] We do not claim here that the reasoning methods we provide outperform more specialised alignment reasoning methods, say for DDL, or alignment debugging: our main contribution is the provision of a unifying framework that works simultaneously at the various levels.

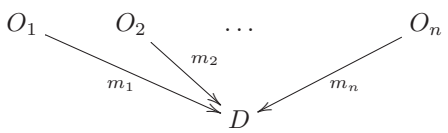
The general picture is then as follows: existing ontologies can be integrated as-is into the DOL framework. With our new extended DOL syntax, we can specify different kinds of alingments. From such an alignment, we construct a graph of ontologies and morphisms between them—in a way depending on the chosen alignment framework. Sometimes, this step also involves transformations on the ontologies, such as relativisation of the (global) domain using predicates. A network of alignments can then be combined to an integrated alignment ontology via a so-called colimit. Reasoning in a network of aligned ontologies is then the same as reasoning in the combined ontology. Thus, in order to implement a reasoner, it is in principle sufficient to define the relativisation procedure for the local logics and the alignment transformation for each kinds of semantics.

3 Networks of ontologies and their semantics

In this section we recall networks of ontologies and their semantics introduced in [23, 8]. Networks of ontologies (here denoted NeO) [8], called distributed systems in [23], consist of a family $(O_i)_{i\in I}$ of ontologies over a set of indexes I interconnected by a set of alignments $(A_{ij})_{i,j\in I}$ between them. Alignments are sets of *correspondences* between the target ontology O_1 and source ontology O_2 of the alignment. Correspondences are triples (e_1, e_2, R) where e_1 and e_2 are entities built with the help of an entity language over O_1 and O_2, respectively, and R is a relation between entities from a set of relations $\mathfrak{R}$.

A semantics of networks of ontologies is given in terms of local interpretation of the ontologies and alignments it consists of. To be able to give such a semantics, one needs to give an interpretation of the relations between entities that are expressed in the correspondences. In the following three subsections let $S = \{(O_i)_{i\in I}, (A_{ij})_{i,j\in I}\}$ be a NeO over a set of indexes I.

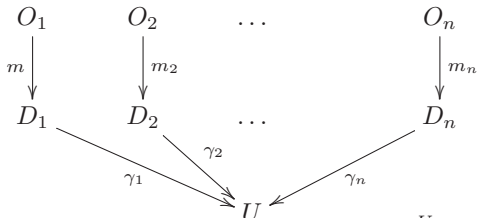
Simple semantics In the simple semantics, the assumption is that all ontologies are interpreted over the same domain (or universe of interpretation) D. The relations in $\mathfrak{R}$ are interpreted as relations over D, and we denote the interpretation of $R \in \mathfrak{R}$ by R^D.



If O_1, O_2 are two ontologies and $c = (e_1, e_2, R)$ is a correspondence between O_1 and O_2, we say that c is satisfied by interpretations m_1, m_2 of O_1, O_2 iff $m_1(e_1)\ R^D\ m_2(e_2)$. This is written $m_1, m_2 \models^S c$. A model of an alignment A between ontologies O_1 and O_2 is then a pair m_1, m_2 of interpretations of O_1, O_2 such that for all $c \in A$, $m_1, m_2 \models^S c$. We denote this by $m_1, m_2 \models^S A$. An interpretation of S is a family $(m_i)_{i\in I}$ of models m_i of O_i. A simple interpretation of S is an interpretation $(m_i)_{i\in I}$ of S over the same domain D.

Definition 1. *[23] A simple model of a S is a simple interpretation $(m_i)_{i\in I}$ of S such that for each $i, j \in I$, $m_i, m_j \models^S A_{ij}$. This is written $(m_i)_{i\in I} \models^S S$. We denote by $Mod^{sim}(S)$ the class of all simple models of S.*

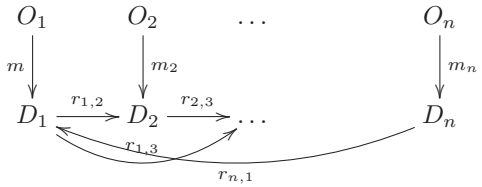
Integrated Semantics Another possibility is to consider that the domain of interpretation of the ontologies of a NeO is not constrained, and a global domain of interpretation U exists, together with a family of *equalising functions* $\gamma_i : D_i \to U$, where D_i is the domain of O_i, for each $i \in I$. A relation R in $\mathfrak{R}$ is interpreted as a relation R^U on the global domain. Satisfaction of a correspondence $c = (e_1, e_2, R)$ by two models m_1 of O_1 and m_2 of O_2 means that $\gamma_i(m_i(e_1))R^U\gamma_j(m_j(e_2))$. We denote this by $m_1, m_2 \models^I_{\gamma_1,\gamma_2} c$ and by $m_1, m_2 \models^I_{\gamma_1,\gamma_2} A$ we denote that $m_1, m_2 \models^I_{\gamma_1,\gamma_2} c$ for each $c \in A$.



An integrated interpretation of S is then $\{(m_i)_{i\in I}, (\gamma_i)_{i\in I}\}$ where $(m_i)_{i\in I}$ is an interpretation of S and $\gamma_i : D_i \to U$ is a function to a common global domain U for each $i \in I$. We here assume that the γ_i are inclusions.[2]

Definition 2. *[23] An integrated interpretation $\{(m_i), (\gamma_i)\}$ of S is an integrated model of S iff for each $i, j \in I$, $m_i, m_j \models^I_{\gamma_i,\gamma_j} A_{ij}$. We denote by $Mod^{int}(S)$ the class of all integrated models of a NeO S.*

Contextualised Semantics The functional notion of contextualised semantics in [23] is not very useful and has been replaced by a more flexible relational notion subsequently [8], closely related to the semantics of DDLs [2] and $\mathcal{E}$-connections [14].



The idea is to relate the domains of the ontologies by a family of relations $r = (r_{ij})_{i,j\in I}$. The relations R in $\mathfrak{R}$ are interpreted in each domain of the ontologies in the NeO. Satisfaction of a correspondence $c = (e_1, e_2, R)$ by two models m_1 of O_1 and m_2 of O_2 means that $m_i(e_1)R^i r_{ji}(m_j(e_2))$, where R^i is the interpretation of R in D_i. We denote it by $m_1, m_2 \models^C_r c$, and extend this to alignments, denoted $m_1, m_2 \models^C_r A$ if all correspondences of the alignment are satisfied by m_1, m_2 w.r.t. r.

A contextualised interpretation of S is a pair $\{(m_i)_{i\in I}, (r_{ij})_{i,j\in I}\}$ where $(m_i)_{i\in I}$ is an interpretation of S and $(r_{ij})_{i,j\in I}$ is a family of domain relations such that r_{ij} relates the domain of m_i to the domain of m_j and r_{ii} is the identity (diagonal) relation. Further assumptions about domain relations can be added, thus restricting more the class of interpretations of a NeO.

Definition 3. *A contextualised model of the NeO S is a contextualised interpretation $((m_i)_{i\in I}, (r_{ij})_{i,j\in I})$ of S such that for each $i, j \in I$, $m_i, m_j \models^C_r A_{ij}$. We denote by $Mod^{con}(S)$ the class of all contextualised models of a NeO S.*

[2] The theory also works for injections without much change. Arbitrary, i.e. possibly non-injective maps, are conceptually not necessary: a local model can be quotiented by the kernel of a non-injective such map, and then be replaced by the quotient, leading to an injective map again.

4 DOL Alignments

In this section we start by introducing the DOL concepts necessary for giving semantics of alignments. We then introduce the syntax of alignments in DOL and illustrate with the help of an example involving OWL ontologies how the semantics of alignments can be given using diagrams and colimits. We then present the main result of the paper, showing how the categorical semantics of DOL alignments captures the three semantics of networks of ontologies.

4.1 DOL Diagrams and Combinations

The syntax for specifying diagrams in DOL is

`graph D =` $D_1, \ldots, D_m, O_1, \ldots, O_n, M_1, \ldots, M_p, A_1, \ldots, A_k$

where D_i are (sub-)diagrams, O_i are ontologies, M_i are morphisms and A_i are alignments. The user specifies a diagram D formed with the subgraphs given by diagrams D_i, extended with ontologies O_i and the morphisms M_i and the subdiagrams of the alignments A_i

DOL also provides means for combining a diagram of ontologies into a new ontology, such that the symbols related in the diagram are identified. The syntax of combinations is `ontology O = combine D`, where D is a diagram, named or specified as above. The semantics of a combination O is the class of models of the colimit ontology of the diagram specified in the combination. Under rather mild technical assumptions, this model class captures exactly the models of the diagram.

4.2 Syntax of DOL Alignments

DOL represents the general alignment format in a similar way to the Alignment API [7] as follows:

`alignment A :` O_1 `to` O_2 `=`
  s_1^1 REL^1 s_2^1, ..., s_1^n REL^n s_2^n
  `assuming DOMAIN`
`end`

where O_1 and O_2 are the ontologies to be aligned, s_1^i and s_2^i are O_1 and respectively O_2 symbols, for $i = 1, \ldots, n$, s_1^i REL^i s_2^i is a *correspondence* which identifies a relation between the ontology symbols, using one of the symbols $>$ (subsumes), $<$ (is subsumed), $=$ (equivalent), $\%$ (incompatible), $\in$ (instance) or $\ni$ (has instance) and `DOMAIN` records whether single, integrated or contextualised semantics is used, using the constant `SingleDomain, GlobalDomain` and `ContextualisedDomain` respectively.

Before starting to analyse the three semantics for NeOs in our setting, we can first define the diagram of a NeO in terms of the diagrams of its parts.

Definition 4. *The diagram of a NeO* $S = \{(O_i)_{i \in I}, (A_{ij})_{i,j \in I}\}$ *is obtained by putting together the diagrams of all alignments* A_{ij} *it consists of.*

The gap to be filled is the construction of the diagram associated with a single assignment, in all three possible assumptions about the semantics. Once this has been given, we can define the semantics of a NeO as the colimit ontology of its associated diagram.

Example 1. We illustrate the three approaches to semantics with the help of a simple example. Let us consider the following two ontologies:

ontology S = **Class:** *Person*
Individual: *alex* **Types:** *Person*
Class: *Child*

ontology T = **Class:** *HumanBeing*
Class: *Male* **SubClassOf:** *HumanBeing*
Class: *Employee*

together with the following correspondences: *S:Person* = *T:HumanBeing*, *S:alex* ∈ *T:Male* and *S:Child* ⊑ ¬ *T: Employee*.

Using the AlignmentAPI syntax, we can write this alignment as

alignment A : S **to** T = *Person* = *HumanBeing*,
alex ∈ *Male*,
Child < ¬ *Employee*

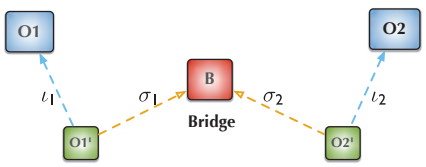
The assumption about the domains of S and T, which determines which of the three semantics is used, is left to be added in the specification of *A*.

In all three cases, the semantics of the alignment is the class of models of the colimit of the diagram of the alignment, which can be specified in DOL by writing `ontology C = combine A`.

4.3 Simple Semantics

In this simplest case, we simply turn the correspondences into OWL sentences to generate the bridge ontology. Moreover, for each entity occuring in an alignment we want to use both its axiomatisation in the original ontology as well as the bridge axioms introduced by the alignment. For this reason, we keep track of the dependency between the symbols of the bridge ontology and the ontology they have origin from by adding a common source in the diagram for these two occurences. This is a well-known construction, see [24].

Definition 5. *Let A be an alignment (using the notations of Sec. 4.2). The diagram of the alignment is of the following shape (a W-alignment in the sense of [24]):*

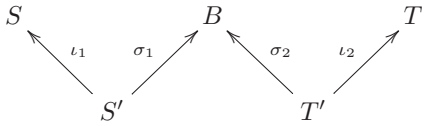


Its constituents are obtained as follows. The ontologies O'_1 and O'_2 collect, respectively, all the symbols s_1 and s_2 that appear in a correspondence $s_1 REL\ s_2$ in A, and have no sentences. The morphisms ι_i from O'_i to O_i, where $i = 1, 2$, are inclusions. The ontology B is constructed by turning the correspondences of the alignment into OWL *axioms. The morphisms σ_1 and σ_2 map the symbols occurring in correspondences to their counterpart in B. The alignment is ill-formed when it contains an equivalence between symbols of different kinds, or if B fails to be a well-formed ontology.*

Example 2. We start by adding the assumption that we have a shared domain for the ontologies in the alignment of Ex. 1:

alignment A : S **to** T = ...
assuming SingleDomain

The diagram of A is then



where S' consists of the concepts *Person* and *Child* and the individual *alex* and T' consists of the concepts *HumanBeing*, *Employee* and *Male*, ι_1 and ι_2 are inclusions and σ_1 and σ_2 map, respectively, *Person* and *HumanBeing* to *Person_HumanBeing* and all other concepts and/or individuals identically.
The bridge ontology B is:

ontology B = **Class:** *Person_HumanBeing*
Class: *Employee*
Class: *Male*
Class: *Child* **SubClassOf:** ¬ *Employee*
Individual: *alex* **Types:** *Male*

The colimit ontology of the diagram of A is:

ontology C = **Class:** *Person_HumanBeing*
Class: *Employee*
Class: *Male* **SubClassOf:** *Person_HumanBeing*
Class: *Child* **SubClassOf:** ¬ *Employee*
Individual: *alex* **Types:** *Male*, *Person_HumanBeing*

4.4 Integrated Semantics

Capturing integrated semantics in DOL using families of models compatible with a diagram is more difficult, as compatibility with the diagram implies uniqueness of the domain. To remedy this, we use relativisation of an ontology where the universal concept becomes a new concept and thus can be interpreted as a subset of the relativised domain. Relativisations have previously been used in defining Common Logic modules [19] or in the re-encoding of DDL into OWL [6].

Definition 6. *Let O be an* OWL *ontology. We define the* relativisation *of O, denoted $\tilde{O}$, as follows. The concepts of $\tilde{O}$ are the concepts of O together with a new concept, denoted $\top_O$. The roles and individuals of $\tilde{O}$ are the same as in O. $\tilde{O}$ contains axioms stating that*

– *each concept C of O is subsumed by $\top_O$,*
– *each individual i of O is an instance of $\top_O$,*
– *each role r has its domain and range, if present, intersected with $\top_O$, otherwise they are $\top_O$.*

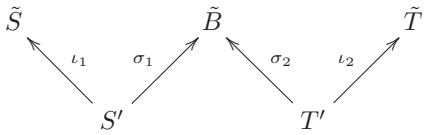
and the axioms of O where the following replacement of concepts is made:

– *each occurence of $\top$ is replaced by $\top_O$, and*
– *each concept $\neg C$ is replaced by $\top_O \sqcap \neg C$*
– *each concept $\forall R.C$ is replaced by $\top_O \sqcap \forall R.C$.*

Example 3. We add the assumption that we have a global domain where the domains of the ontologies in our alignment are included:

alignment A : S **to** T = ...
assuming GlobalDomain

The diagram of A is then



where S' consists of the concepts $Things_S$, *Person* and *Child* and the individual *alex* and T' consists of the concepts $Thing_T$, *HumanBeing*, *Employee* and *Male*, ι_1 and ι_2 are inclusions and σ_1 and σ_2 map *Person* and respectively *HumanBeing* to *Person_HumanBeing* and all other concepts and/or individuals identically.

The relativisations $\tilde{S}$ and $\tilde{T}$ of the ontologies S and T are

ontology $\tilde{S}$ = **Class:** $Things_S$
Class: *Person* **SubClassOf:** $Things_S$
Individual: *alex* **Types:** *Person*, $Things_S$
Class: *Child* **SubClassOf:** $Things_S$

ontology $\tilde{T}$ = **Class:** $Thing_T$
Class: *HumanBeing* **SubClassOf:** $Thing_T$
Class: *Male* **SubClassOf:** *HumanBeing*, $Thing_T$
Class: *Employee* **SubClassOf:** $Thing_T$

The relativised bridge ontology of an alignment is built by relativising the axioms that result from translating the correspondences of A to OWL sentences. Since we made the assumption that equalising functions are all inclusions, there is no need to introduce explicit symbols for them in the bridge ontology. In our case, the bridge ontology of A is

ontology $\tilde{B}$ = **Class:** $Things_S$ **Class:** $Thing_T$
Class: *Person_HumanBeing* **SubClassOf:** $Things_S$, $Thing_T$
Class: *Male* **Class:** *Employee*
Class: *Child* **SubClassOf:** $Thing_T$ **and** ¬ *Employee*
Individual: *alex* **Types:** *Male*

The colimit ontology of the relativised diagram of the alignment in Ex. 1 is:

ontology C = **Class:** *ThingS*
Class: *ThingT*
Class: *Person_HumanBeing* **SubClassOf:** *ThingS*, *ThingC*
Class: *Male* **SubClassOf:** *Person_HumanBeing*
Class: *Employee* **SubClassOf:** *ThingT*
Class: *Child* **SubClassOf:** *ThingS*
Class: *Child* **SubClassOf:** *ThingT* **and** $\neg$ *Employee*
Individual: *alex* **Types:** *Male*, *Person_HumanBeing*

4.5 Contextualised Semantics

Here we need to introduce explicitly the relations between the domains in the language of the bridge ontology. The diagram of the alignment has thus the same shape as in Def. 5, but now the bridge ontology is computed differently and, as in the previous section, the ontologies are relativised. We denote the bridge ontology by $\overline{B}$ and define it to modify B as follows:

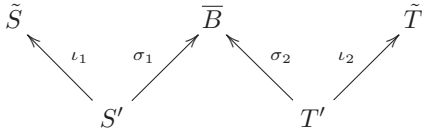
- r_{ji} is added to $\overline{B}$ as a role with domain $\top_T$ and range $\top_S$
- the correspondences are translated to axioms involving these roles:
  - $C_i = C_j$ becomes $C_i \equiv \exists r_{ji} \bullet C_j$
  - $a_i = a_j$ becomes $a_i\ r_{ji}\ a_j$
  - $a_i \in C_j$ becomes $a_i \in \exists r_{ji} \bullet C_j$
  - $C_i < C_j$ becomes $C_i \sqsubseteq \exists r_{ji} \bullet C_j$
  - $C_i \% C_j$ becomes $C_i \sqcap \exists r_{ji} \bullet C_j = \emptyset$
- the properties of the r_{ji} are added as axioms in $\overline{B}$.

Here we assume that the alignment A_{ij} contains no correspondence (r_i, r_j, R), where r_i and r_j are roles. Having such correspondences leads to sentences that cannot be expressed in OWL.

Example 4. We add the assumption that we have different domains for the ontologies, which are related by domain relations:

alignment A : S **to** T = ...
assuming ContextualisedDomain

The diagram of A is then



where the constituents of the diagram, except $\overline{B}$, are as defined in Ex. 3. The bridge ontology of A now becomes:

ontology $\overline{B}$ = **Class:** *ThingS*
Class: *ThingT*
ObjectPropery: r_{TS} **Domain:** *ThingT* **Range:** *ThingS*
Class: *Person* **EquivalentTo:** r_{TS} **some** *HumanBeing*
Class: *Employee*
Class: *Male*
Class: *Child* **SubClassOf:** r_{TS} **some** $\neg$ *Employee*
Individual: *alex* **Types:** r_{TS} **some** *Male*

The colimit ontology of this diagram is:

ontology C = **Class:** *ThingS*
Class: *ThingT*
ObjectPropery: r_{TS} **Domain:** *ThingT* **Range:** *ThingS*
Class: *Person* **EquivalentTo:** r_{TS} **some** *HumanBeing*
Class: *Male* **SubClassOf:** *Person_HumanBeing*
Class: *Employee*
Class: *Child* **SubClassOf:** r_{TS} **some** ¬ *Employee*
Individual: *alex* **Types:** r_{TS} **some** *Male*, *Person*

4.6 The three semantics in DOL

In this section let $S = ((O_i)_{i\in I}, (A_{ij})_{i,j\in I})$ be a network of OWL ontologies. We denote $C(S)$ the colimit ontology of the diagram associated to S, regardless if the assumption about the alignments in S is that they use single, integrated or contextualised semantics. The model class of $C(S)$ is denoted $[\![C(S)]\!]$.

Theorem 1. *1. If the alignments of S use* `SingleDomain` *and the diagram of S is connected, then $[\![C(S)]\!]$ is in bijection with $Mod^{sim}(S)$.*
2. If the alignments of S use `GlobalDomain`*, then $[\![C(S)]\!]$ is in bijection with the class $Mod^{int}(S)$ of integrated models $((m_i),(\gamma_i))$ of S where γ_i are inclusions.*
3. If the alignments of S use `ContextualisedDomain`*, then $[\![C(S)]\!]$ is in bijection with $Mod^{con}(S)$.*

DOL is supported by Ontohub (`https://ontohub.org`), a Web-based repository engine for managing distributed heterogenous ontologies. The back-end of Ontohub is the Heterogeneous Tool Set HETS [18] which is used for parsing, static analysis and proof management of ontologies. HETS supports alignments and combinations: it generates the diagram of an alignment according to the assumption on the domain and can compute colimits of OWL ontologies automatically.

5 Conclusions and Future Work

Our theoretical contributions to the foundations of ontology alignment and combination have a potentially large impact on future alignment practices and reasoning. Regardless of the semantic paradigm employed, 'reasoning' with alignments involves at least three levels: (1) the finding/discovery of alignments (often based heavily on statistical methods), (2) the construction of the aligned ontology (the 'colimit'), and (3) reasoning over the aligned result, respectively debugging and repair, closing the loop to (1). Our contributions in this paper address levels (2) and (3).

Regarding (2), platforms such as Bioportal (with hundred thousands of mappings) illustrate that mappings between ontologies, ontology modules, and the concepts and definitions living in them, are of great importance to support re-use. The importance of alignment has also been well demonstrated for foundational ontologies in the repository ROMULUS [13]. In the case of Bioportal, the DOL language allows to declaratively manage sets of alignments, and to give precise semantics. In the case of ROMULUS, it allows to align ontologies such as Dolce or BFO expressed in first-order logic with OWL versions of the same ontology.

Regarding (3), alignment tools such as LogMap [12] and ALCOMO [16] employ reasoning over aligned ontologies and repair either parts of the input ontologies or revise the mappings (one technique to enable this is to re-encode the mappings into a global OWL ontology) to restore global consistency. Using DOL and the reasoning capabilities of the Hets/Ontohub ecosystem, such tools could be used to directly operate on a NeO, and to update the diagram structure accordingly.

The approach presented here provides an integration of the major paradigms of ontology alignment in one coherent framework. This includes standard alignment relations, DDLs, PD-L, IDDL, and $\mathcal{E}$-connections [14] which we currently study in more detail. Our construction assumes OWL as the local logic of the ontologies; however it can be generalised to an arbitrary logic by giving a (necessary logic-specific) relativisation procedure and alignment transformation. Moreover, DOL's support for heterogeneity allows us not only to handle heterogeneous alignment, but also to move to a more expressive logic when a bridge axiom cannot be expressed in the local logic of the ontologies. Thus we can remove the restriction on correspondences in the contextualised semantics.

Future work includes the combination of different alignment paradigms within one network (as principally enabled by our unifying framework) and an integration of techniques for the revision of NeOs [9] into DOL. In our setting, the propagation of detected repairs into a network could be done by updating the alignment mappings and re-computing the alignment diagrams. Further work is also needed for the problem of reasoning about the consequences of a NeO; here we expect module extraction to provide an increase in performance of proof search. At the tool level, the integration of the three semantics for alignments in Ontohub is currently in progress. Ontohub is already compatible with the OWL API, and its potential for interoperability is increased further by the integration of the Alignment API.

Acknowledgements We gratefully acknowledge the financial support of the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under FET-Open Grant number: 611553, project COINVENT.

References

1. J. Bao, G. Voutsadakis, G. Slutzki, and V. Honavar, 'Package-based description logics', in *Modular Ontologies*, 349–371, Springer, (2009).
2. A. Borgida and L. Serafini, 'Distributed Description Logics: Assimilating Information from Peer Sources', *Journal of Data Semantics*, **1**, 153–184, (2003).
3. P. Bouquet, F. Giunchiglia, F. van Harmelen, L. Serafini, and H. Stuckenschmidt, 'C-OWL: Contextualizing ontologies', in *ISWC*, pp. 164–179, (2003).
4. D. Calvanese, G. De Giacomo, and M. Lenzerini, 'Description logics for information integration', in *Computational Logic: Logic Programming and Beyond*, eds., A. C. Kakas and F. Sadri, volume LNCS 2408, pp. 41–60. Springer, (2002).
5. D. Calvanese, G. De Giacomo, M. Lenzerini, and R. Rosati, 'Logical foundations of peer-to-peer data integration', in *PODS*, eds., C. Beeri and A. Deutsch, pp. 241–251. ACM, (2004).
6. B. Cuenca-Grau and O. Kutz, 'Modular Ontology Languages Revisited', in *Proc. of the IJCAI'07 Workshop on Semantic Web for Collaborative Knowledge Acquisition (SWeCKa), Hyderabad, India, January 2007*, pp. 22–31, (2007).

7. J. David, J. Euzenat, F. Scharffe, and C. Trojahn dos Santos, 'The alignment API 4.0', *Semantic Web*, **2**(1), 3–10, (2011).
8. J. Euzenat and P. Shvaiko, *Ontology matching*, Springer-Verlag, Heidelberg (DE), 2nd edn., 2013.
9. Jérôme Euzenat, 'Foundations for revising networks of ontologies', in *Third International Workshop on Debugging Ontologies and Ontology Mappings-WoDOOM14*, p. 1, (2014).
10. C. Ghidini and L. Serafini, 'Distributed first order logics', in *Frontiers Of Combining Systems 2, Studies in Logic and Computation*, pp. 121–140. Research Studies Press, (1998).
11. P. Hitzler, J. Euzenat, M. Krötzsch, L. Serafini, H. Stuckenschmidt, H. Wache, and A. Zimmermann, 'Integrated view and comparison of alignment semantics', Contrat, (December 2005).
12. E. Jiménez-Ruiz and B. Cuenca Grau, 'LogMap: Logic-based and Scalable Ontology Matching', in *Proc. of the 10th International Semantic Web Confernece (ISWC 2011)*, 273–288, Springer, (2011).
13. Z. Khan and M. Keet, 'The foundational ontology library ROMULUS', in *Model and Data Engineering*, 200–211, Springer, (2013).
14. O. Kutz, C. Lutz, F. Wolter, and M. Zakharyaschev, '$\mathcal{E}$-Connections of Abstract Description Systems', *Artificial Intelligence*, **156**(1), 1–73, (2004).
15. O. Kutz, T. Mossakowski, and M. Codescu, 'Shapes of alignments - construction, combination, and computation', in *International Workshop on Ontologies: Reasoning and Modularity (WORM-08)*, eds., U. Sattler and A. Tamilin, volume 348 of *CEUR-WS online proceedings*, (2008).
16. C. Meilicke, H. Stuckenschmidt, and A. Tamilin, 'Repairing ontology mappings', in *AAAI*, volume 3, p. 6, (2007).
17. T. Mossakowski, O. Kutz, M. Codescu, and C. Lange, 'The Distributed Ontology, Modeling and Specification Language', in *Proceedings of the 7th International Workshop on Modular Ontologies (WoMO-13)*, ed., C. Del Vescovo et al., volume 1081. CEUR-WS, (2013).
18. Till Mossakowski, Christian Maeder, and Klaus Lüttich, 'The Heterogeneous Tool Set', in *TACAS 2007*, eds., Orna Grumberg and Michael Huth, volume 4424 of *Lecture Notes in Computer Science*, pp. 519–522. Springer-Verlag Heidelberg, (2007).
19. F. Neuhaus and P. Hayes, 'Common Logic and the Horatio problem', *Applied Ontology*, **7**(2), 211–231, (2012).
20. P. Shvaiko and J. Euzenat, 'Ontology matching: State of the art and future challenges', *IEEE Trans. Knowl. Data Eng.*, **25**(1), 158–176, (2013).
21. A. Zimmermann, 'Integrated distributed description logics', in *Description Logics*, eds., D. Calvanese, E. Franconi, V. Haarslev, D. Lembo, B. Motik, A. Turhan, and S. Tessaris, volume 250 of *CEUR Workshop Proceedings*, (2007).
22. A. Zimmermann, 'Logical formalisms for agreement technologies', in *Agreement Technologies*, ed., S. Ossowski, volume 8 of *Law, Governance and Technology Series*, 69–82, Springer Netherlands, (2013).
23. A. Zimmermann and J. Euzenat, 'Three semantics for distributed systems and their relations with alignment composition', in *Proc. 5th International Semantic Web Conference (ISWC)*, LNCS 4273, pp. 16–29, Athens (GA US), (2006).
24. A. Zimmermann, M. Krötzsch, J. Euzenat, and P. Hitzler, 'Formalizing Ontology Alignment and its Operations with Category Theory', in *Proc. of FOIS-06*, pp. 277–288, (2006).

The Properties of Property Alignment

Michelle Cheatham and Pascal Hitzler

Data Semantics (DaSe) Laboratory, Wright State University, Dayton OH 45435, USA

Abstract. The performance of alignment systems on property matching lags significantly behind that on class and instance matching. This work seeks to understand the reasons for this and consider possible avenues for improvement. The paper contains an in-depth exploration of the performance of current alignment systems on the only commonly accepted alignment benchmark that involves matches between properties. A second benchmark involving properties is also proposed. Finally, an entirely string-based approach targeted towards aligning properties is presented and evaluated using both benchmarks.

1 Introduction

Previously, we conducted an analysis of the performance of string similarity metrics on ontology alignment tasks [2]. One of the findings of that work was that string metrics perform much worse on properties than on classes. Commonly used preprocessing strategies such as stopword removal or consideration of synonyms using WordNet were ineffective at improving performance.

Others have noted the challenge of aligning properties as well. For example, this is stated without additional detail by Maedche and Staab in [7], while Pernelle et al. note that human experts had a more difficult time agreeing on when properties match than on when classes do [11]. In this paper we build on previous work by considering the difference in performance of full-featured alignment systems on properties versus classes (Section 2). In additional to overall performance, we look at the false positives and false negatives commonly made by current systems when aligning properties within the OAEI Conference track. Then, because the Conference track is the *only* commonly used non-synthetic alignment benchmark that involves properties, we introduce a potential new benchmark to allow for verification of results in Section 3. In Section 4 we continue our exploration of the limits of string-centric approaches for ontology alignment by introducing an entirely string-based property alignment system and evaluating its results on both the Conference track and our newly-proposed secondary benchmark. The results compare favorably to the best-performing string similarity metric and PARIS, a full-featured alignment system.

2 OAEI Conference Track

The OAEI Conference track is the only established non-synthetic test set for alignment systems that has reference alignments containing matches between

System	Class Prec	Class Rec	Class Fms	Prop Prec	Prop Rec	Prop Fms
AML	0.86	0.62	0.72	1.00	0.20	0.33
AMLback	0.86	0.64	0.73	1.00	0.24	0.39
CIDER_CL	0.46	0.59	0.52	0.07	0.22	0.11
HerTUDA	0.84	0.56	0.67	0.26	0.20	0.23
HotMatch	0.81	0.57	0.67	0.24	0.20	0.22
IAMA	0.87	0.55	0.67	0.14	0.07	0.09
LogMap	0.82	0.65	0.73	0.62	0.28	0.39
MapSSS	0.74	0.59	0.66	0.00	0.00	0.00
ODGOMS	0.87	0.55	0.67	0.32	0.26	0.29
ODGOMS1_2	0.81	0.66	0.73	0.32	0.26	0.29
ServOMap_v104	0.74	0.65	0.69	0.00	0.00	0.00
StringsAuto	0.71	0.63	0.67	0.00	0.00	0.00
WeSeEMatch	0.85	0.54	0.66	0.50	0.02	0.04
WikiMatch	0.84	0.54	0.66	0.26	0.22	0.24
YAM++	0.82	0.71	0.76	0.68	0.57	0.62
Average	0.79	0.60	0.68	0.36	0.18	0.21

Table 1. Performance of the top 2013 OAEI competitors on classes versus properties

properties as well as classes. Table 1 shows the results of the top 2013 OAEI competitors on the Conference track, broken down into classes and properties[1]. The average f-measure for classes is more than three times that for properties.

Table 2 presents the most common correct and incorrect property matches identified by the participants in the 2013 OAEI, along with the valid property matches that were most frequently omitted by those systems. The frequency column in the table indicates the number of alignment systems out of the 15 qualifying[2] systems that produced (or failed to produce, in the case of false negatives) each match. The first section of the table shows that the equivalent properties that were most frequently correctly identified all have very high string similarity. Unfortunately, the second section shows that high string similarity is also the defining characteristic of the most common false positives. It may seem surprising that some of the matches in this section of the table are not valid. In some cases the domain or range of the matched properties indicate that they are not being used in the same way. For instance, the domain of cmt:name is the union of Person and Conference whereas the domain of sigkdd:Name is only Person and a separate property, Name_of_conference, is used to represent a conference's name. In other cases the match may make sense in isolation but would lead to logical inconsistency of the merged ontology. Finally, we see in the last section of the table that the properties involved in the most common false negatives generally have a much lower string similarity, such as cmt:hasBeenAssigned and ekaw:ReviewerOfPaper. In many of these cases, the domain and range of the properties *do* have strong syntactic similarity however, e.g. Reviewer and Paper for hasBeenAssigned and Possible_Reviewer and Paper for reviewerOfPaper. Further, there were some quite frequently missed equivalent properties that have

[1] MapSSS and StringsAuto do not attempt to align properties
[2] Those performing better than the basic edit-distance string metric

strong clues in the labels themselves, such as cmt:writePaper and confOf:writes. Of the 31 common false negatives, 13 have noticeable string similarity.

3 YAGO-DBPedia

In addition to the OAEI Conference test set, we would also like to analyze the performance of alignment systems on properties from another real-world alignment task. For this we have chosen DBPedia[3] and YAGO.[4] DBPedia is a linked data version of the information in Wikipedia. The YAGO knowledge base has been automatically extracted from Wikipedia, WordNet, and GeoNames by researchers at the Max Planck Institute for Computer Science. Both DBPedia and YAGO contain millions of instances and thousands of schema-level entities. This scale is too large for many current alignment systems. We are specifically interested in aligning the properties of these two datasets, so we have extracted a cohesive subset of each one that will allow us to do this without requiring an inordinately long runtime. This was done using the following procedure:

1. For each property in YAGO, randomly choose five facts that involve the property. For properties with less than five facts, use all that are available.
2. Add the classes (type) of every instance mentioned in the facts from step 1.
3. Randomly add up to five other facts related to the instances from step 1.
4. Repeat step 2 for any additional instances added during step 3.
5. Compute the "closure" of this set of entities by recursively retrieving all schema-related axioms related to any entity within our sample.

The procedure for creating the DBPedia sample was analogous, except that instead of randomly choosing the facts in step 1, we selected facts with the same instances as our YAGO sample when available. This is possible because, since DBPedia and YAGO both represent information from Wikipedia, there is error-free mapping of instances that point to the same Wikipedia page. When there were no matching YAGO instances for the facts related to a particular DBPedia property, we reverted to randomly choosing facts. The characteristics of these dataset samples are shown in Table 3.

This dataset sample may be of use to other researchers, so we have made it publicly available at http://www.michellecheatham.com/files/dbpedia-yago.zip. It should be noted that DBPedia and YAGO have some idiosyncrasies. For instance, many properties defined in the ontologies are never used or are incompletely defined (e.g. missing domain or range definitions). Also, the definitions of some properties are spread across a datatype property, which specifies the range, and an annotation property, which specifies the domain. Furthermore, some of the properties appear to be used inconsistently, or at least more broadly than they are defined. For instance, in DBPedia we see that the instance HAL 9000

[3] http://wiki.dbpedia.org/Downloads39

[4] http://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/

Type	Property 1	Property 2	Freq.
Correct	cmt:email	confOf:hasEmail	11
	confOf:hasFirstName	edas:hasFirstName	11
	conference:has_an_email	confOf:hasEmail	9
	cmt:email	conference:has_an_email	9
	conference:has_the_last_name	edas:hasLastName	9
	conference:has_a_review	ekaw:hasReview	9
	conference:has_the_first_name	edas:hasFirstName	9
	conference:has_the_first_name	confOf:hasFirstName	9
False Positive	iasted:pay	sigkdd:pay	9
	confOf:hasEmail	edas:hasEmail	9
	cmt:email	edas:hasEmail	8
	cmt:name	sigkdd:Name	8
	confOf:hasPhone	edas:hasPhone	8
	confOf:hasStreet	edas:hasStreet	7
	confOf:hasPostalCode	edas:hasPostalCode	7
	iasted:obtain	sigkdd:obtain	7
	confOf:hasTopic	edas:hasTopic	7
	conference:has_an_email	edas:hasEmail	7
	cmt:writtenBy	confOf:writtenBy	7
False Negative	cmt:hasBeenAssigned	ekaw:reviewerOfPaper	15
	cmt:assignExternalReviewer	conference:invites_co-reviewers	15
	cmt:assignedByReviewer	conference:invited_by	15
	edas:endDate	sigkdd:End_of_conference	15
	conference:is_given_by	sigkdd:presentationed_by	15
	conference:has_a...tutorial_topic	confOf:hasTopic	15
	conference:contributes	iasted:write	15
	cmt:hasBeenAssigned	confOf:reviewes	15
	conference:gives_presentations	sigkdd:presentation	15
	conference:has_the_last_name	confOf:hasSurname	15
	cmt:assignedTo	ekaw:hasReviewer	15
	confOf:reviewes	edas:isReviewing	15
	confOf:hasSurname	edas:hasLastName	15
	conference:has_a_review_expertise	edas:hasRating	15
	cmt:writtenBy	ekaw:reviewWrittenBy	15
	cmt:hasSubjectArea	confOf:dealsWith	14
	cmt:writePaper	confOf:writes	14
	edas:isReviewedBy	ekaw:hasReviewer	14
	cmt:hasAuthor	confOf:writtenBy	14
	confOf:writes	edas:hasRelatedPaper	14
	edas:hasCostAmount	sigkdd:Price	14
	cmt:assignedTo	edas:isReviewedBy	14
	edas:startDate	sigkdd:Start_of_conference	14
	cmt:hasConferenceMember	edas:hasMember	14
	cmt:hasBeenAssigned	edas:isReviewing	14
	edas:hasLocation	ekaw:heldIn	14
	edas:hasName	sigkdd:Name_of_conference	14
	edas:isReviewing	ekaw:reviewerOfPaper	14
	confOf:hasEmail	sigkdd:E-mail	13
	conference:has_an_email	sigkdd:E-mail	13
	conference:contributes	ekaw:authorOf	13

Table 2. Most common correct, false positive, and false negative property matches identified by alignment systems in the 2013 OAEI

Dataset	DBPedia	YAGO
Classes	617	10962
Object Properties	1046	85
Data Properties	1407	37
Named Individuals	8685	1680
Datatypes	23	23
Annotations	77	125
Total Entities	11855	12912

Table 3. Characteristics of the DBPedia and YAGO samples

has a gender property with a value of male and that Eaglet (Alice's Adventures in Wonderland) has a gender value female. In some cases the gender property is used differently, however: the instance Alexander has a gender property value of Alexandra, and the value for Maine North High School is mixed-sex education. While these issues can be a pain to work with, they are realistic concerns that ontology alignment systems will need to face for many application scenarios.

There is currently no curated alignment of the properties in the DBPedia and YAGO datasets. We would like to use the crowdsourcing approach described in Section 4 based on Amazon's Mechanical Turk system to create a complete reference alignment for the properties in these two datasets. It is not realistic to crowdsource opinion on all possible pairs of properties, however. A set of potential mappings is needed to bootstrap the crowdsourcing effort. Unfortunately, not many alignment systems have made results available for this pair of ontologies. The developers of the PARIS alignment system are the exception – they have produced and made public a set of subsumption relationships between properties [14]. We can consider the cases where subsumption relations between two properties exist in both directions as indicative of an equivalence relation. We will use these matches, together with those produced by a basic string similarity metric and by our string-based property matcher described in the next section to begin the process of crowdsourcing a viable reference alignment. Due to the limited number of alignment approaches providing the potential matches to verify, this method will allow us to assess precision reasonably well but recall values are likely to be less accurate. While less than ideal, this is a common method of evaluation in the absence of an established reference alignment [6,14,12]. Mechanical Turk has previously been successfully used by other researchers for a similar purpose – verifying relationships within biomedical ontologies [9].

4 String-based Property Alignment

In this section we present an entirely string-based approach to property alignment, which we will call PropString.[5]

Four strings are extracted for each property: the label, the core concept, the domain, and the range. The label is simply the entity's label. The core

[5] http://michellecheatham.com/files/PropString.zip

concept is either the first verb in the label that is greater than four characters long or, if there is no such verb, the first noun in the label, together with any adjectives that modify that noun. For example, the label "wrote paper" has the core concept "wrote" and the label "has corresponding author" has the core concept "corresponding author." We arrived at this technique through an analysis of common naming patterns for properties. We used the Standford log-linear part of speech tagger to compute the core concept [18]. The domain (resp. range) string is a concatenation of the labels of any classes in the domain (resp. range) of a property. The similarity of each of these four pairs of strings is then computed using the Soft TF-IDF metric, which was the string metric shown in [2] to have the best performance on properties.

While the vast majority of alignment systems use a string similarity metric, they use them in different ways. One approach is to find highly precise "anchor" matches which serve as the seed that the rest of the alignment grows out from. Another approach is to use a string metric to filter out any obviously incorrect matches in order to reduce computational complexity. This requires a string metric with high recall. To address both of these use cases, the PropString approach can be run in two configurations: precision-oriented and recall-oriented. In the precision-oriented mode, a pair of entities is considered a match if the similarity values for their core concepts, domains, **and** ranges are all greater than the threshold. In the recall-oriented mode, the pair is considered a match if the similarity values for their core concepts **or** their domains and ranges are greater than the threshold.

Allowing matches based solely on high similarity of domain and range in the recall-oriented configuration results in very low precision unless further steps are taken. We use a combination of two approaches to reduce the number of false positives. The first is the calculation of the confidence value: this is done by averaging the similarity values for the exact labels, their domains, and their ranges. The second is that we keep a list of each entity that is considered a match so far, along with the entity it maps to and the confidence value. Every time a new potential match between properties is identified, its confidence value is checked against any existing current matches involving those properties. If the new match has a greater confidence value, the old match is removed in favor of the new one, otherwise the new match is ignored. Using the exact label similarity when computing the confidence values rather than the core concept eliminates the loss of precision associated with extracting the core concept, effectively breaking any ties in favor of the closer lexical matches. The effect of this approach is that any properties with the same domain and range act as a filter, with the specific match from that set chosen based on the actual property label.

4.1 Evaluation: Conference track

Table 4 shows the results of PropString on the OAEI Conference track. The system was configured with a threshold of 0.9 and to only include matches in which both entities were in the namespace of the ontologies to be matched (in accordance with the OAEI guidelines). The results are compared with those of Soft

System	Precision	Recall	F-measure
PropString (prec)	1.0	0.26	0.41
PropString (rec)	0.34	0.5	0.4
Soft TF-IDF	0.2	0.24	0.22

Table 4. Results on the OAEI Conference track

TF-IDF with a threshold of 0.8. This was shown in [2] to be the best-performing string metric for property alignment. It is evident that PropString greatly outperforms Soft TF-IDF on this test set. The precision-oriented configuration of PropString quintuples the precision of Soft TF-IDF (to a perfect 1.0) while maintaining roughly the same recall. Analogously, the recall-oriented version doubles the recall of Soft TF-IDF while still achieving noticeably better precision. The f-measures for both the precision- and recall-oriented configurations are double that of Soft TF-IDF.

We also conducted a series of tests which show that there are no redundant aspects to the PropString metric: removing any element reduces performance. In particular, removing the idea of extracting the core concept from property labels has such a disastrous effect on recall that the precision-oriented configuration becomes useless. Similarly, removing either the best match filter or using simple label similarity for the confidence value rather than averaging label, domain, and range similarity cuts precision in half in the recall-oriented configuration. Consideration of domain and range in the similarity computation is shown to be the key to this approach.

4.2 Evaluation: YAGO-DBPedia

We also evaluated the performance of PropString on the YAGO-DBPedia alignment task. We compare the performance of PropString to that of the basic Soft TF-IDF similarity metric and the PARIS alignment system. PARIS is an acronym for Probabilistic Alignment of Relations, Instances, and Schema. The system approaches property alignment by considering the degree of overlap between the sets of instances involving each property [15].

There is no established reference alignment for the DBPedia and YAGO ontologies. We begin the process of creating one by collecting the equivalent property relationships generated by PropString, Soft TF-IDF, and PARIS and using Amazon's Mechanical Turk to verify their accuracy. In total, these three approaches produced 133 unique equivalence matches that involved properties. We formulated questions for each match of the form "Does property label A mean the same thing as property label B?" Respondents were instructed to choose one of four options: they mean the same thing, one is a more general or more specific term than the other, they are related in some other way, or there is no relation. We provided these more nuanced options rather than just yes or no because we would like to eventually develop a reference alignment useful for evaluating the performance of alignment systems that produce all types of matches. There has been some debate in the alignment community on how to

phrase questions to crowdsourcing participants in order to acquire good-quality matches [13]. In order to provide some context, we provided information about the domain and range of each property and up to five examples of instances with values for each property.

The 133 matches were grouped into 19 sets of 7 questions each, and we paid 25 cents for each set. Preliminary testing showed that the general response on these nuanced verification questions were not very reliable (others have indicated problems with scammers for these tasks as well [9]). We therefore invited only Turkers who had previously demonstrated good performance on alignment verification tasks to participate in this one. There were ten of these individuals, and we received input from 6 or 7 of them for each match.

Rather than requiring precise agreement on the type of relationship (if any) for each potential match, it might make sense for our current purposes to consider a weaker sense of agreement. One option is to consider two answers to be in agreement if they both either indicate some relationship exists or they both conclude there is no relation between the two properties. In this case, if one person indicated two entities are related in a sub/super relationship and another indicated that they are equivalent, these answers would be considered in agreement. Two answers would only be seen to disagree if one indicated there is no relation at all and the other disagreed. This way of interpreting the results might be useful for an alignment system if the results from this phase were being used to either find all types of relationships between entities or to gather all possible matches and use further processing to filter the set down to only equivalence relations. We will call this "recall-oriented." Figure 1 (top) shows the results of PARIS, Soft TF-IDF, and PropString on the YAGO-DBPedia property alignment task using this definition of correctness.

Another possible way to interpret the results is to consider two answers to be in agreement only if they both conclude either that the entities are precisely equivalent or that they are not equivalent. Using this viewpoint, if one person indicated that two entities are related in a sub/super relationship and another indicated that they are precisely equivalent, these answers would be seen as disagreeing. If instead one person considered the match to be a sub/super relationship and another considered them to have no relationship at all, these two individuals would be seen as in agreement because they both conclude that there is no equivalence relationship. This interpretation may be useful if an alignment system is attempting to find high-quality equivalence relations between entities, which it may subsequently use as a seed for further processing. We will refer to this as "precision-oriented." Figure 1 (bottom) is analogous to Figure 1 (top) but uses this precision-oriented definition of correctness.

The basic string metric Soft TF-IDF produces the highest precision, regardless of how correctness is measured. Further, that precision is 0.79 and .96 (depending on evaluation approach), which is on par with the degree of agreement among the Turkers on these matches. So we see that a straightforward string metric can in some ways outperform more sophisticated alignment strategies. In fact, PARIS and the precision-based configuration of PropString have such low

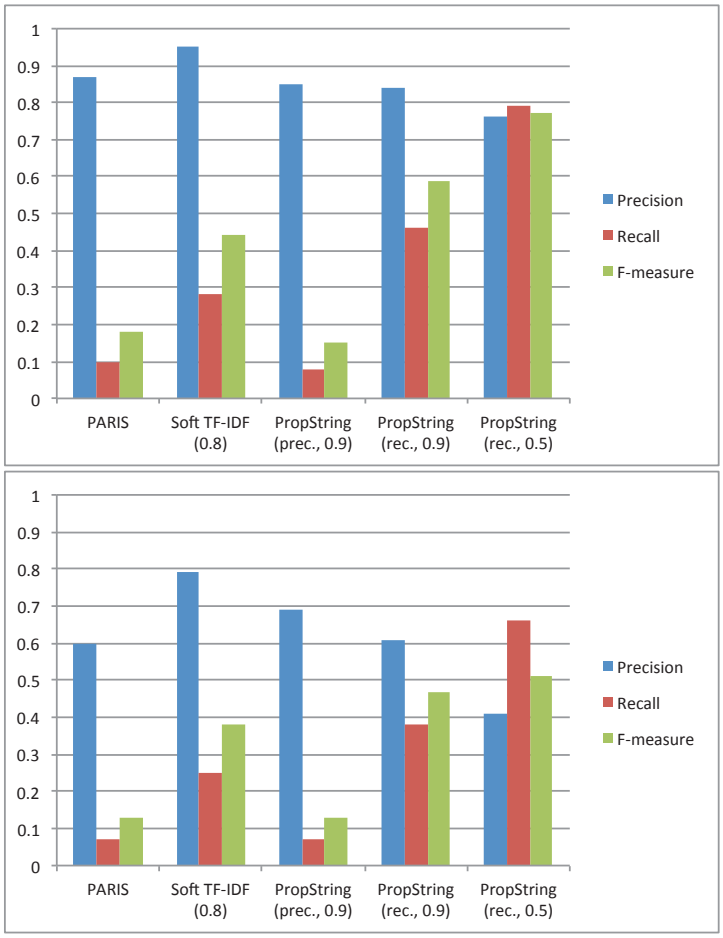


Fig. 1. Results of the YAGO-DBPedia alignment task: Recall-oriented evaluation (top), precision-oriented evaluation (bottom)

recall that they may not be of much utility for many application scenarios. This is surprising considering the strong performance of this PropString configuration on the properties within the Conference track.

Another thing to note from these results is the very strong performance of the recall-based configuration of PropString, both relative to the other approaches and in an absolute sense. When PropString is run in its recall configuration with a threshold of 0.5, both the precision and recall are in the neighborhood of that produced by much more complex alignment systems on the simpler task of class equivalence in smaller test sets, such as the Conference track. Of course, the very preliminary nature of the YAGO-DBPedia reference alignment must be kept in

mind. More work, hopefully involving results produced by many other alignment system on this pair of ontologies, is needed to confirm these results.

5 Related Work

The only existing work focused specifically on property matching of which we are aware is the extensional approach described in [3] and the pattern-based approach to finding complex mappings (i.e. matches involving more than two entities) across ontologies that involve properties explored in [12]. However, there is a large body of existing research on ontology alignment in general. A good survey of current approaches can be found in [1].

There has also been some analysis of the particular characteristics of properties versus classes and instances in ontologies. For instance, [17] discusses common part-of-speech naming patterns for different entity types. Situations in which properties are often reified were considered in [10]. Additionally, it has been shown that taxonomies of properties are much less common than those of classes [17,10] and that some ontologies are class-centric while others are property-centric (e.g. SeasonTicketHolder versus holdsSeasonTicket) [16]. These characteristics may impact the performance on alignment systems on property matching.

In 2002 Melnik and his colleagues developed a strategy called "similarity flooding" to improve the performance of alignment systems [8]. This was adapted for ontology alignment by the developers of RiMOM [5]. The basic idea is that an initial pass is made through the datasets to establish a set of high precision anchor mappings, such as exact string matches. Then similarity values are propagated to adjacent nodes. If the similarity value of two nodes reaches a threshold, they are considered equivalent. This technique may improve the performance on property alignment by leveraging the increased accuracy of class and instance alignment. Suchanek et. al. recently applied this ontology-oriented similarity flooding approach in their PARIS alignment system, which identifies both equivalences and subsumptions for classes and properties [14]. They found that while class alignments did not facilitate alignment of properties or instances, there was significant interplay between the latter two. This was particularly true for functional or nearly functional properties, in which any domain value maps to only one range value.

There have been several attempts to modify the standard similarity flooding approach to further improve the performance on property matching. For example, comparison of instance data and datatype property range values can be improved by using different similarity metrics for strings, dates, integers, etc. [20]. Further, in deference to the difficulty of matching properties, it is possible to propagate a fraction of the normal similarity values when adjacent properties are compatible rather than definite matches. This is the approach taken in [11] where compatibility for properties is defined as those with domains and ranges that are either the same or subtypes of one another.

The PropString algorithm's consideration of the lexical similarity of the domain and range of properties is somewhat similar to the work by Vizenor and his colleagues in [19]. Their approach, which is focused on the biomedical domain, used domain and range similarity as a sanity check on the alignment of properties. PropString's extraction of the "core concept" within property labels based on parts of speech is somewhat related to more general NLP mapping approaches, such as that found in [4].

6 Conclusions and Future Work

This work explored the performance of current ontology alignment systems on property alignment using the OAEI Conference track as a benchmark. In addition, a second benchmark involving property matches was suggested. The paper also introduced PropString, an entirely string-based approach to aligning properties. The performance of PropString was evaluated using both benchmarks and was shown to be better than the best-performing string metric by a wide margin. PropString also compared favorably to the PARIS alignment system on the secondary benchmark, based on a crowdsourced evaluation of matches using Mechanical Turk. While the performance of PropString is encouraging, the f-measure on property still lags that of classes, and more work needs to be done in this area.

Several aspects of the work presented here require further validation. In particular, additional experimentation regarding crowdsourcing reference alignments using Mechanical Turk needs to be done to verify the potential uses of the approach. For instance, our preliminary results showed that general users can often give good input on "yes or no" alignment verification tasks but that more complex questions regarding the type of relationship between two entities (e.g. equivalence, subsumption, inverse properties) is more difficult. It would be useful to develop guidelines for when and how to qualify users for different types of alignment tasks. More work in particular remains to be done in order to generate an established high-quality reference alignment for the YAGO-DBPedia alignment task. In order to do this, we need to generate results on this ontology pair using more alignment systems. These results can then be manually verified, either through Mechanical Turk or by experts. Additionally, we would like to incorporate the PropString approach into a full-featured alignment system and evaluate the difference in performance.

Acknowledgments. This work was supported by the National Science Foundation award 1017225 "III: Small: TROn—Tractable Reasoning with Ontologies."

References

1. Bernstein, P.A., Madhavan, J., Rahm, E.: Generic schema matching, ten years later. Proceedings of the VLDB Endowment 4(11), 695–701 (2011)
2. Cheatham, M., Hitzler, P.: String similarity metrics for ontology alignment. In: The Semantic Web–ISWC 2013, pp. 294–309. Springer (2013)

3. Gunaratna, K., Thirunarayan, K., Jain, P., Sheth, A., Wijeratne, S.: A statistical and schema independent approach to identify equivalent properties on linked data. In: Proceedings of the 9th International Conference on Semantic Systems. pp. 33–40. ACM (2013)
4. Klinkmüller, C., Weber, I., Mendling, J., Leopold, H., Ludwig, A.: Increasing recall of process model matching by improved activity label matching. In: Business Process Management, pp. 211–218. Springer (2013)
5. Li, J., Tang, J., Li, Y., Luo, Q.: Rimom: A dynamic multistrategy ontology alignment framework. Knowledge and Data Engineering, IEEE Transactions on 21(8), 1218–1232 (2009)
6. Madhavan, J., Bernstein, P.A., Rahm, E.: Generic schema matching with cupid. In: VLDB. vol. 1, pp. 49–58 (2001)
7. Maedche, A., Staab, S.: Measuring similarity between ontologies. In: Knowledge Engineering and Knowledge Management: Ontologies and the Semantic Web, pp. 251–263. Springer (2002)
8. Melnik, S., Garcia-Molina, H., Rahm, E.: Similarity flooding: A versatile graph matching algorithm and its application to schema matching. In: Proc. 18th ICDE Conf.(Best Student Paper award) (2002)
9. Mortensen, J.M., Musen, M.A., Noy, N.F.: Crowdsourcing the verification of relationships in biomedical ontologies. In: AMIA Annual Symposium Proceedings. vol. 2013, pp. 1020–9. American Medical Informatics Association (2013)
10. Noy, N.F., Hafner, C.D.: The state of the art in ontology design: A survey and comparative review. AI magazine 18(3), 53 (1997)
11. Pernelle, N., Saïs, F., Safar, B., Koutraki, M., Ghosh, T.: N2r-part: identity link discovery using partially aligned ontologies. In: Proceedings of the 2nd International Workshop on Open Data. p. 6. ACM (2013)
12. 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)
13. Sagi, T., Gal, A.: In schema matching, even experts are human: Towards expert sourcing in schema matching. In: 2014 IEEE 30th International Conference on Data Engineering Workshops (ICDEW). pp. 45–49. IEEE (2014)
14. Suchanek, F., Abiteboul, S., Senellart, P.: Ontology alignment at the instance and schema level. arXiv preprint arXiv:1105.5516 (2011)
15. Suchanek, F.M., Abiteboul, S., Senellart, P.: Paris: Probabilistic alignment of relations, instances, and schema. Proceedings of the VLDB Endowment 5(3), 157–168 (2011)
16. Šváb, O.: Exploiting patterns in ontology mapping. In: The Semantic Web, pp. 956–960. Springer (2007)
17. Svátek, V., Šváb-Zamazal, O., Presutti, V.: Ontology naming pattern sauce for (human and computer) gourmets. In: Workshop on Ontology Patterns. pp. 171–178 (2009)
18. Toutanova, K., Klein, D., Manning, C.D., Singer, Y.: Feature-rich part-of-speech tagging with a cyclic dependency network. In: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1. pp. 173–180. Association for Computational Linguistics (2003)
19. Vizenor, L.T., Bodenreider, O., McCray, A.T.: Auditing associative relations across two knowledge sources. Journal of Biomedical Informatics 42(3), 426–439 (2009)
20. Zhao, L., Ichise, R.: Instance-based ontological knowledge acquisition. In: The Semantic Web: Semantics and Big Data, pp. 155–169. Springer (2013)

Completeness and Optimality in Ontology Alignment Debugging

Jan Noessner[1], Heiner Stuckenschmidt[1],
Christian Meilicke[1], and Mathias Niepert[2]

[1] University of Mannheim, 68163 Mannheim, Germany
firstname@informatik.uni-mannheim.de
[2] University of Washington, Seattle, WA 98195-2350, USA
mniepert@cs.washington.edu

Abstract. The benefit of light-weight reasoning in ontology matching has been recognized by a number of researchers resulting in alignment repair systems such as Alcomo and LogMap. While the general benefit of logical reasoning has been shown in principle, there is no systematic empirical evaluation analyzing (i) the impact of completeness of the reasoning methods and (ii) whether approximate or optimal solutions to the conflict resolution problem have to be preferred. Using standard benchmark data sets, we show that increasing the expressive power does improve the matching results and that optimal resolution methods slightly outperform approximate ones.

Keywords: ontology matching, expressiveness, alignment debugging

1 Introduction

Research in ontology matching has been strongly influenced by earlier results in schema matching [19]. There are several approaches that aim at being universally applicable across ontologies and database schemas by relying on a representation of ontologies and schemas as directed graphs [1]. While various studies have verified the benefit of explicit, logical schema semantics such as description logics and logical reasoning (e.g. [17]), there is only a limited number of approaches that exploit schema semantics to improve matching results in a principled manner. Early approaches exploiting the logical structure of class descriptions were based on specialized similarity measures that take logical operators into account (e.g. [2]). Additional methods avoid structural properties that mimic unwanted reasoning results [6] or require user interaction [18]. More recently, a number of approaches have been proposed that explicitly use ontological reasoning. Meilicke et al., for instance, compute and leverage logical inconsistencies to eliminate conflicts between alignment hypotheses [11]. A related approach was proposed by Jiménez-Ruiz et al. [7]. Additional debugging strategies remove incoherent alignments during a post-processing step [20, 13]. Giunchiglia and colleagues use reasoning over logic-based representations of class labels but solely focus on the

problem of matching class hierarchies [4]. Most of these approaches exploit restricted forms of reasoning so as to ensure the scalability to large models. While these approaches demonstrated the benefits of logical reasoning for matching expressive ontologies, there has not been a *systematic* investigation of the impact logical reasoning has on matching results. In particular, it is not obvious whether more expressive reasoning methods provide more benefits than less expressive ones. Furthermore, the impact of applying different strategies for resolving detected logical conflicts, has not been analyzed in details. Within this paper we report about experiments that shed light on both research questions. Another systematic evaluation is provided in [8], where the authors focus on the need of debugging and provide a comparison of two debugging systems, while we focus on completeness and optimality.

The paper is structured as follows. In Section 2 we explain alignment incoherence and introduce the notion of completeness and optimality with respect to alignment debugging. Moreover, we describe three existing debugging systems that we use in our experiments. We discuss the setting of our experiments in Section 3 with a focus on data sets and evaluation metrics. The results of these experiments are presented in Section 4. We close with a discussion in Section 5.

2 Incoherence in Ontology Matching

Ontology Matching is the task of finding correspondences between entities of two ontologies $\mathcal{O}_1$ and $\mathcal{O}_2$. According to [3], a correspondence between an entity e_1 defined in $\mathcal{O}_1$ and an entity e_2 defined in $\mathcal{O}_2$ is a 4-tuple $\langle e_1, e_2, r, n\rangle$ where r is a semantic relation (such as equivalence), and n is a real-valued confidence value. A set of correspondences is called an alignment. In line with most matching systems and benchmarks, we focus on equivalence correspondences, i.e., $(\langle e_1, e_2, \equiv, n\rangle)$, where the matched entities are either both classes or properties. However, the overall approach can also be applied to any kind of axioms as long as these axioms are supported by the debugging system (e.g., all three systems used in our experiments support also subsumption axioms as correspondences).

An alignment $\mathcal{A}$ can be created by a human expert or by an automated matching system. In both cases, $\mathcal{A}$ might include erroneous correspondences. However, it is reasonable to assume that $\mathcal{O}_1$ and $\mathcal{O}_2$ do not contain erroneous axioms. For that reason, an alignment $\mathcal{A}$ can be interpreted as a set of uncertain, weighted equivalence axioms, while $\mathcal{O}_1 \cup \mathcal{O}_2$ will comprise the certain axioms. Merging $\mathcal{A}$, $\mathcal{O}_1$, and $\mathcal{O}_2$ can then result into an incoherent ontology, i.e. some of the classes of $\mathcal{O}_1$ or $\mathcal{O}_1$ might be unsatisfiable due to the additional information encoded in $\mathcal{A}$. The following example shows an incoherent alignment.

$$\mathcal{O}_1 = \{\texttt{Jaguar}_1 \sqsubseteq \texttt{Cat}_1, \texttt{Cat}_1 \sqsubseteq \texttt{Animal}_1\},$$
$$\mathcal{O}_2 = \{\texttt{Jaguar}_2 \sqsubseteq \texttt{Brand}_2, \texttt{Animal}_2 \sqsubseteq \neg\texttt{Brand}_2\}$$
$$\mathcal{A} = \{\langle \texttt{Jaguar}_1 \equiv \texttt{Jaguar}_2, 0.9\rangle, \langle \texttt{Animal}_1 \equiv \texttt{Animal}_2, 0.95\rangle\}$$

In this example the classes $\texttt{Jaguar}_1$ and $\texttt{Jaguar}_2$ are unsatisfiable in the merged ontology. There are three possible ways to resolve this incoherence: (1) Dis-

card both correspondences, (2) discard $\langle \texttt{Jaguar}_1 \equiv \texttt{Jaguar}_2, 0.9 \rangle$, or (3) discard $\langle \texttt{Animal}_1 \equiv \texttt{Animal}_2, 0.95 \rangle$. Obviously, we prefer (2) and (3) over (1). Moreover, it seems to make more sense to remove the correspondence that is less confident, i.e., the most reasonable decision is (2) given no further information is available.

However, with larger matching problems a solution to the debugging problem becomes more complex for two reasons. First, not all conflicts (= subsets of correspondences resulting in incoherence) might be detected. This might be caused by using an incomplete reasoning technique, for example, because only a certain type of axioms are analyzed. Second, the detected conflicts might be overlapping and there are several ways to resolve the incoherence. In such a situation a solution should be preferred that removes as less confidence as possible. We call such a solution an optimal solution and define it as a subset $\Delta \subseteq \mathcal{A}$ such that $\mathcal{A} \setminus \Delta$ is coherent and there exist no other Δ^* such that $\mathcal{A} \setminus \Delta^*$ is coherent and $\sum_{c \in \Delta} conf(c) > \sum_{c \in \Delta^*} conf(c)$. This definition corresponds to the definition of a global optimal diagnosis given in [9].

Note that optimality and completeness are independent characteristics of a debugging system. It is possible to construct a debugging system that is complete in terms of reasoning but cannot guarantee the optimality of the solution, while it is also possible to construct a system that is incomplete and optimal, in the sense that the solution is optimal with respect to all detected conflicts, even though these conflicts are only a subset of all conflicts due to the incompleteness. Note also that the notion of optimality is a technical notion, i.e., an optimal solution might not always be the best solution in terms of precision and recall.

3 Experimental Set-Up

3.1 Datasets

The ontologies we use for the experiments are from the ontology alignment evaluation initiative (OAEI) [5]. We selected the CONFERENCE and the LARGE BIOMED ontologies because these benchmarks are not artificially created (unlike, for instance, the BENCHMARKS dataset), are not focused on a narrow alignment problem (unlike, for instance, the MULTIFARM dataset which is concerned with multilingual ontology matching), and provide coherent reference alignments. Moreover, the size of the LARGE BIOMED ontologies allows us to assess the scalability of the presented approach.

The CONFERENCE dataset consists of 15 ontologies which model the domain of conference organization [21]. The number of classes, properties, and axioms of a particular type of 7 ontologies are listed in Table 1 ordered by increasing expressiveness. Every row in the table, with the exception of the last row, corresponds to one expressiveness level we used for the experiments (see Section 4.1). For the 7 listed ontologies, reference alignments were created for each possible pair, resulting in 21 ontology pairs with a reference alignment.

Since the ontologies in the CONFERENCE dataset are relatively small, we also performed experiments with the LARGE BIOMED ontologies. The corresponding

	cmt	conf.	confof	edas	ekaw	iasted	sigkdd
classes	36	60	38	104	77	140	49
properties	59	64	36	50	33	41	28
subsumption	25	49	33	84	71	132	41
+ disjointness	52	63	76	491	145	133	41
+ domain and range restrictions	149	149	100	543	184	193	73
+ all other $\mathcal{EL}^{++}$ axioms	263	331	293	865	309	505	186
every axiom	318	408	335	903	341	539	193

Table 1. Number of classes, properties, and axioms of the CONFERENCE ontologies.

data set consists of the Foundational Model of Anatomy (FMA)[3], National Cancer Institute Thesaurus (NCI)[4], and SNOMED clinical terms[5] ontologies. Semantically rich and with thousands of classes, the problem of aligning these ontologies is one of the computationally most challenging in the OAEI campaign. For the 2013 OAEI campaign, only 12 out of 21 participating system configurations were able to compute results for the three combinations. We used the "small fragment" matching problems of the track. For more details on these data sets we refer the reader to the OAEI track website[6]. The properties of the ontologies are summarized in Table 2.

3.2 Alignment Aggregation

For each of the matching tasks described above, there are several alignments available that have been generated by different matching system. We decided to aggregate these alignment for each matching task in a preprocessing step. Thus, we can work with large input alignments and can avoid an additional subsequent aggregation of the debugging results. We aggregated the results of all matchers participating in the 2013 OAEI campaign. For the CONFERENCE benchmark, we included all matchers which performed better than the string equality baseline [5]. For the LARGE BIOMED benchmark, we included the results of the 6 matchers which were able to compute a solution for every combination [5]. The participants in the LARGE BIOMED track were allowed to submit results for different settings of their system. We always used the best results of each system in terms of f-measure.

The method of alignment aggregation resembles the approach described in [9]. For each pair of ontologies, we union the alignments $\mathcal{A}_1, \ldots, \mathcal{A}_i, \ldots, \mathcal{A}_n$ of each matching system i to one alignment $\mathcal{A}$. To that end, we first span the confidence values w of each correspondence $\langle w, a \rangle$ in alignment $\mathcal{A}_i$ to the range of $(0, 1]$. This ensures that the confidence values of the individual matchers are scaled identically. We then compute the aggregated a-priori confidence values for a

[3] http://sig.biostr.washington.edu/projects/fm/
[4] http://ncit.nci.nih.gov/
[5] http://www.ihtsdo.org/index.php?id=545
[6] http://www.cs.ox.ac.uk/isg/projects/SEALS/oaei/2013/

	FMA_{NCI}	$\text{FMA}_{\text{SNOMED}}$	NCI_{FMA}	$\text{NCI}_{\text{SNOMED}}$	$\text{SNOMED}_{\text{FMA}}$	$\text{SNOMED}_{\text{NCI}}$
classes	3696	10157	6488	23958	13412	51128
properties	24	24	63	82	18	51
subsumption	3693	10154	4917	18946	16287	31299
+ disjointness	3732	10196	5022	19099	16287	31299
+ domain and range restrictions	3732	10196	5130	19233	16287	31299
+ all other $\mathcal{EL}^{++}$ axioms	7521	20449	14269	50218	33673	122221
every axiom (without annotations)	7548	20478	15634	54452	47104	122221

Table 2. Number of classes, properties, and deterministic axioms of the LARGE BIOMED ontologies. For each ontology there exists two fragments. In case of the FMA ontology, for example, one fragment contains the axioms overlapping with the NCI ontology and one fragment contains the axioms overlapping with the SNOMED ontology.

correspondence as the normalized sum of all a-priori confidences of that correspondence. The average size of one alignment for the CONFERENCE benchmark is 42 ranging from at least 29 to at most 60 correspondences. For the LARGE BIOMED benchmark, we obtain 3396 correspondences for the ontology pair NCI and FMA; 10760 for the pair FMA and SNOMED; and 18842 for SNOMED and NCI.

3.3 Debugging Systems

In our evaluation we present results for the debugging systems ELOG, LOGMAP and ALCOMO that we apply on the ontologies and alignments described so far.

- ELOG [16] is a reasoner for log-linear description logics, which offers complete reasoning capabilities for EL++. ELOG can be used for debugging ontology alignments (details can be found in [15]). Since ELOG transforms the debugging problem to finding the MAP state of a Markov Logic Network, it guarantees the optimality of the solution, i.e., the MAP state corresponds to an optimal solution. However, ELOG is not complete with respect to the full expressiveness of OWL DL.
- LOGMAP [7] is a matching system including a component for alignment debugging. In our experiments we report only about applying this component and refer to it, for the sake of simplicity, as LOGMAP. This component translates the ontologies into a set of Horn clauses and applies the linear Dowling-Gallier algorithm for propositional Horn satisability multiple times for repairing. The algorithm is not optimal and to our knowledge also not complete against the OWL DL profile. LOGMAP is known to be the most efficient debugging tool currently available (see for example [8]).
- ALCOMO [9] has specifically been developed for the purpose of debugging ontology alignments. ALCOMO can be used in a setting that ensures the completeness (for OWL DL) and the optimality of the solution. The optimality of the solution is guaranteed by applying an exhaustive search algorithm to check potential solutions. However, this setting is applicable only to small

matching problems. Using a lightweight setting, ALCOMO can also be applied to larger matching problems loosing both the features of optimality and completeness.

3.4 Metrics

F-Measure Precision and recall of an alignment $\mathcal{A}$ measure the correctness of $\mathcal{A}$ and the completeness of $\mathcal{A}$, respectively. Both measures are defined with respect to a given reference alignment or gold standard $\mathcal{G}$. The F-measure is the harmonic mean of precision an recall. Precision P, recall R, and F-measure F can be formally defined as

$$P = \frac{|\mathcal{A} \cap \mathcal{G}|}{|\mathcal{A}|}, \quad R = \frac{|\mathcal{A} \cap \mathcal{G}|}{|\mathcal{G}|}, \quad \text{and} \quad F = \frac{2 \cdot P \cdot R}{P + R}.$$

Number of Unsatisfiable Classes The number of unsatisfiable classes is proposed as a quality measure for ontology matching in [10]. It refers to the number of classes that are unsatisfiable in the merged ontology $\mathcal{A} \cup \mathcal{O}_1 \cup \mathcal{O}_2$ where $\mathcal{O}_1$ and $\mathcal{O}_2$ are the matched ontologies and $\mathcal{A}$ is the alignment between them. The smaller the number of unsatisfiable classes the higher the quality of the alignment. We computed the number of unsatisfiable classes with the HERMIT [12] reasoner since it is known from previous work [8] that HERMIT outperforms other reasoners in the computation of unsatisfiable classes. Unfortunately, we were not able to compute the unsatisfiable classes for the NCI and SNOMED pair under 5 hours and, thus, cannot provide the number of unsatisfiable classes for the LARGE BIOMED benchmark.

The CONFERENCE benchmark experiments were performed on a virtual machine with 8 GB RAM and 2 cores with 2,4 Ghz. The LARGE BIOMED experiments were executed on a virtual machine with 60 GB RAM and 2 cores.

4 Experimental Results

4.1 Expressiveness

Within this section we report about experiments that include axiom types with increasing expressiveness. Within these experiments we use the ELOG debugging system. For the lowest level of expressiveness, we only include subsumption axioms $A \sqsubseteq B$. For the second level, we add disjointness axioms $A \sqcap B \sqsubseteq \bot$. For the third level, we include domain and range restrictions. Finally, for the most expressive level, we include all axioms representable with the DL $\mathcal{EL}^{++}$. The size of the resulting ontologies is shown in Table 1 and 2 presented in the previous section. The ELOG debugging system is complete with respect to each of the resulting matching problems. However, with this approach we simulate different types of debugging systems that are restricted to exploit different levels of expressiveness. For example, on the second level we simulate a debugging

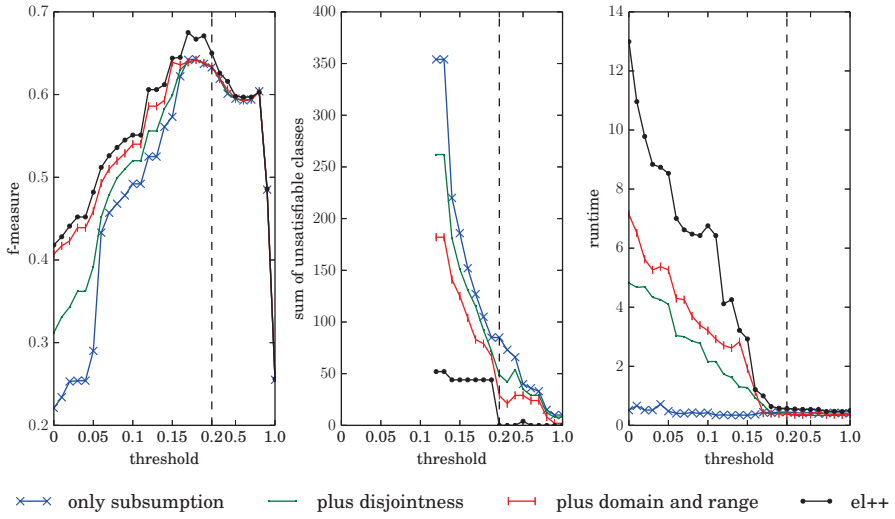


Fig. 1. Results for the CONFERENCE benchmark. With increasing expressiveness, F-measure and runtime (in seconds) are increasing. Contrary, the incoherent classes decrease. This effects are stronger for lower thresholds since more conflicts occur. For thresholds lower than 0.12 the HERMIT reasoner failed in computing the number of incoherent classes. In total, the conference benchmark contains 2.973 classes.

system that bases its reasoning techniques only on the inter-dependencies between subsumption and disjointness axioms. Note that we analyze results for the ontologies in their full expressiveness in the subsequent section.

Figure 1 and Figure 2 depict the results for the various levels of expressiveness and for different thresholds for the CONFERENCE and LARGE BIOMED benchmarks, respectively. The x-axis shows the different thresholds that we applied prior to the debugging step. The results show that the differences between the various levels are less pronounced for lower thresholds. Hence, we put a special emphasis on the threshold areas below 0.2 (for the CONFERENCE benchmark) and below 0.7 (for the LARGE BIOMED benchmark) since results for higher thresholds were nearly identical. Please note that in Figure 1 the stepsize in each chart changes at threshold 0.2 from 0.01 to 0.1 since, beyond that threshold, there are only very few logical conflicts.

We observe a positive correlation between increased expressiveness and F-measure scores. Considering only subsumption axioms results in lower scores compared to the setting with additional disjointness axioms. Even higher F-measures scores are achieved if domain and range restrictions are taken into account. The highest F-measure scores are obtained if we incorporate all $\mathcal{EL}^{++}$ axioms. This holds also true for the choice of a well-suited threshold in the range

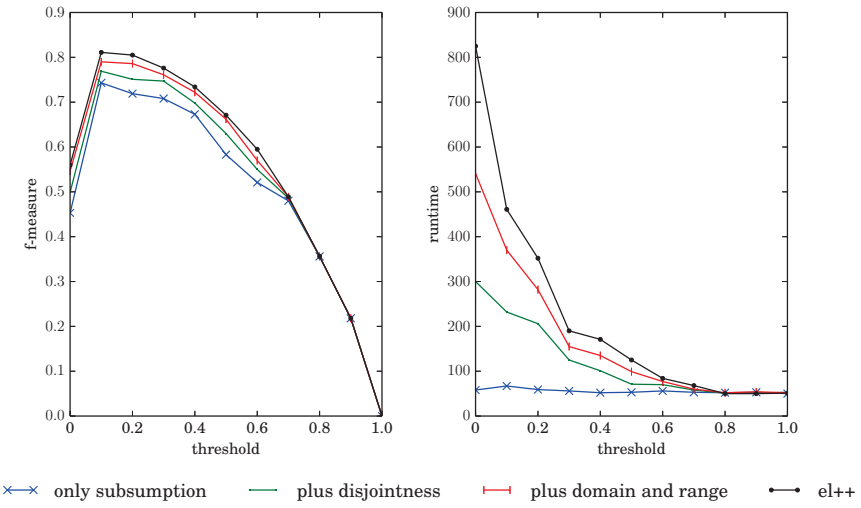


Fig. 2. Results for the LARGE BIOMED benchmark. With increasing expressiveness, F-measure scores and running time (in seconds) are increasing. These effects are stronger for lower thresholds since more conflicts occur. We do not provide the number of incoherent classes because the HERMIT reasoner did not terminate within 5 hours.

of 0.15 to 0.2 in case of the CONFERENCE benchmark, where we clearly observe the benefits of exploiting the full expressiveness of $\mathcal{EL}^{++}$.

As expected, the number of unsatisfiable classes (center figure of Figure 1) is higher for settings with decreased expressiveness. For the subsumption only configuration, we observe the highest number of unsatisfiable classes in the final alignment. On the other hand, there are only few unsatisfiable classes for the $\mathcal{EL}^{++}$ setting. Aside from the F-measure results, this is another indication of an improved alignment quality. The reason why we obtain unsatisfiable classes at all for $\mathcal{EL}^{++}$ expressiveness is that the expressiveness of our underlying ontologies is higher than $\mathcal{EL}^{++}$. In case of the LARGE BIOMED benchmark the HERMIT reasoner was not able to determine the number of unsatisfiable classes within 5 hours. Thus, we do not provide a graphic for this benchmark.

Also as expected, we observe an increase in running time (right figures) when the number of resolved conflicts increases, since runtimes are higher for low thresholds. Furthermore, runtimes also increase with increasing expressiveness. This is in line with our expectation, because a higher level of expressiveness results also in the generation of a more complex optimization problem that needs to be solved when computing the most probable coherent ontology query.

In summary, the results show that the alignment quality increases with an increase in expressiveness. F-measure scores are higher and the number of unsatisfiable classes is lower if expressiveness increases. We can also conclude that

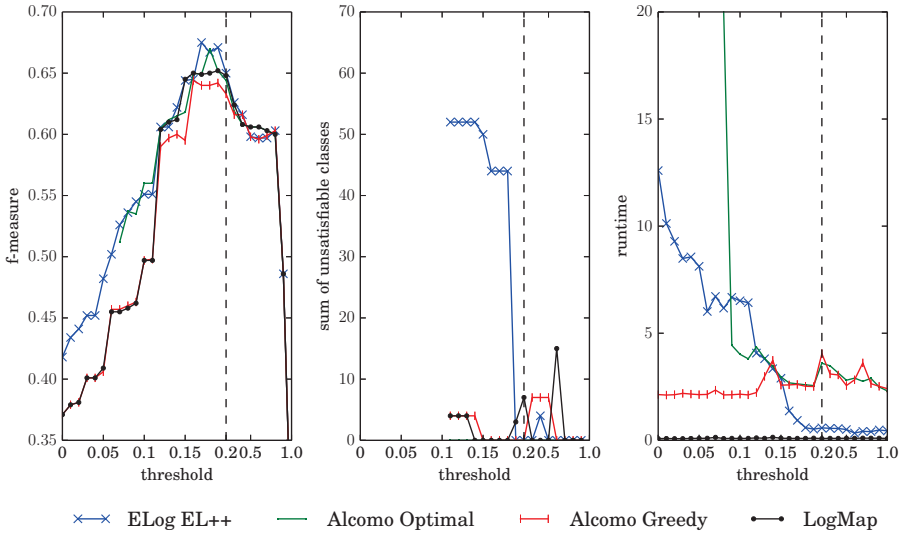


Fig. 3. Results for ELog compared with other approaches on the CONFERENCE benchmark. For lower thresholds, optimal approaches achieve a higher F-measure than approximate approaches but require a longer runtime. The number of unsatisfiable classes is low (1.7% or lower) for all systems. For thresholds lower than 0.12 the HERMIT reasoner failed in computing the number of incoherent classes. In total, the conference benchmark contains 2.973 classes. Runtimes are given in seconds.

a debugging system that is more complete in terms of the supported expressivity will generate better results compared to a less complete system. Runtimes, however, increase with higher expressiveness. This shows a trade-off between runtime and alignment quality depending on the choice of the supported expressiveness.

4.2 Approximate vs. optimal solutions

In this section, we experimentally address the question if optimal algorithms lead to higher quality than approximate algorithms. To that end, we compare the log-linear description logic system ELog and the optimal algorithm of ALCOMO against the approximate algorithms of LOGMAP and ALCOMO.[7]

The results for the CONFERENCE and LARGE BIOMED benchmark are depicted in Figure 3 and Figure 4, respectively. Again, we focus on the discussion

[7] ALCOMO can be executed in different settings. We refer to the setting using the parameters `METHOD_OPTIMAL/REASONING_COMPLETE` as optimal algorithm. We refer to the setting `METHOD_GREEDY/REASONING_EFFICIENT` as approximate algorithm. However, this settings is both incomplete and does not generate an optimal solution.

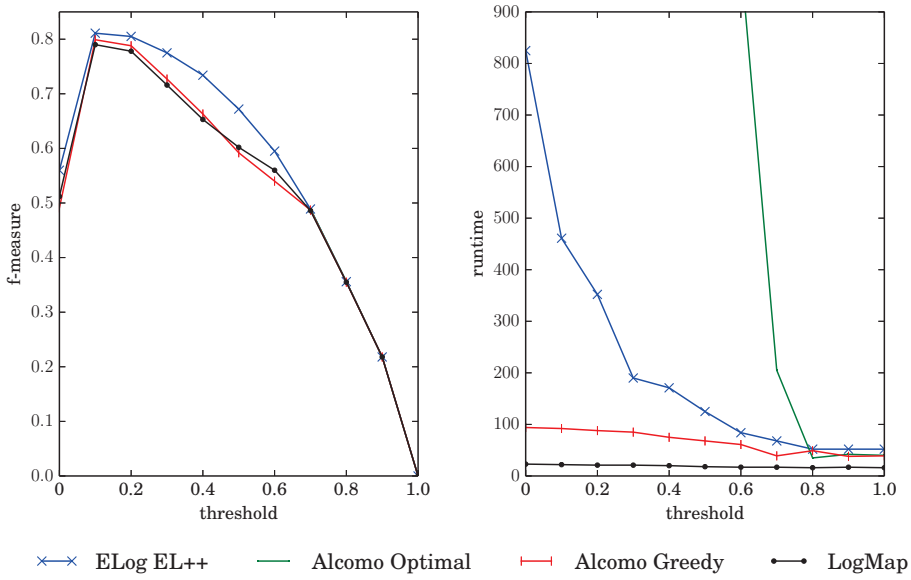


Fig. 4. Results for ELog compared with other approaches on the LARGE BIOMED benchmark. For lower thresholds, optimal approaches achieve a higher F-measure than approximate approaches but require a longer runtime. We do not provide the number of incoherent classes because the HERMIT reasoner did not terminate within 5 hours. Runtimes are given in seconds.

of results for thresholds below 0.2 (for the CONFERENCE benchmark) and 0.7 (for the LARGE BIOMED benchmark).

The system ELog and the optimal algorithm of ALCOMO gains the highest F-measure scores (left figures). The approximate algorithms of ALCOMO and LOGMAP reach lower F-measure scores. The difference in F-measure results between ELog and the optimal algorithm of ALCOMO is due to the fact that the associated optimization problems often have more than one solution. Each of this optimal solution has the same objective, i.e. the confidence total of the resulting alignments is the same, but sometimes different F-measure scores. Thus, ELog might choose a different optimum than the optimal algorithm of ALCOMO.

ELog has the highest number of unsatisfiable classes (center figure of Figure 3) of all three algorithms. However, having 53 inconsistent classes is only 1.7% compared to the total sum of classes of 2,973. As explained above, ELog is complete only for $\mathcal{EL}^{++}$. Thus, all inconsistencies were caused from axioms which are out of the scope of $\mathcal{EL}^{++}$. The results indicate that the restricted expressivity seems to be less important than the optimality of the solution, since ELog generates at the same time results with the best F-measure.

The approximate algorithms of LogMap and Alcomo are more efficient, especially for lower thresholds. In case of the conference benchmark, ELog outperforms the approximate Alcomo algorithm for thresholds higher than 0.15. Except for the thresholds of 0.11 and 0.12, the exact Alcomo algorithm is slower than ELog and does not terminate within one hour for thresholds below 0.09. For the large BioMed benchmark, the approximate algorithms are faster. For thresholds below 0.7 the exact Alcomo algorithm does not terminate within one hour. LogMap achieves by far the best runtime results, which is also supported by the results reported in [8]. This is (at least partially) caused by incomplete reasoning and non-optimal conflict resolution techniques.

The non-optimal variant of Alcomo and LogMap generate very similar alignments. This becomes obvious when comparing the F-measure scores presented in the left plots of Figure 3 and 4. Obviously, the systems show a similar bevaviour and seem to apply a similar conflict resolution strategy. The same observation can be made for the optimal variant of Alcomo and ELog. Thus, the distinction between optimal and non-optimal algorithms becomes visible in the threshold/F-measure plots, which supports the importance of this distinction.

Overall, we can conclude that optimal systems achieve higher F-measure scores than the approximate algorithms. With respect to runtime, the approximate algorithms are faster than the optimal approaches. In particular LogMap outperforms all other systems. Furthermore, ELog has shorter runtimes than the optimal algorithm of Alcomo. This is remarkable since LogMap and Alcomo are specialized on ontology matching. They leverage the fact that weighted axioms can only occur *between* ontologies and that those axioms are either subsumption or equivalence axioms.

5 Conclusions

Our experiments indicate that an increase in expressiveness leads to an increase in F-measure scores. Furthermore, the comparison of approximate and optimal ontology alignment repairing systems shows that optimal approaches achieve better F-measure scores. However, we observe a trade-off between F-measure and runtime. Runtimes are longer for higher expressiveness and optimal approaches have, on average, longer runtimes than approximate approaches. Thus, we advice users to employ optimal approaches for non-time critical data integration tasks. If real-time ontology alignment is required, we recommend the use of approximate approaches combined with reasoning techniques that might be incomplete.

References

1. Aumueller, D., Do, H.H., Massmann, S., Rahm, E.: Schema and ontology matching with coma++. In: Proceedings of the 24th International Conference on Management of Data (SIGMOD). pp. 906–908 (2005)
2. Euzenat, J., Valtchev, P.: Similarity-based ontology alignment in OWL-lite. In: Proceedings of the 16th European Conference on Artificial Intelligence (ECAI). pp. 333–337 (2004)

3. Euzenat, J., Shvaiko, P.: Ontology matching. Springer (2007)
4. Giunchiglia, F., Shvaiko, P.: Semantic matching. Knowledge Eng. Review 18(3), 265–280 (2003)
5. Grau, B.C., Dragisic, Z., Eckert, K., Euzenat, J., Ferrara, A., Granada, R., Ivanova, V., Jiménez-Ruiz, E., Kempf, A.O., Lambrix, P., Nikolov, A., Paulheim, H., Ritze, D., Scharffe, F., Shvaiko, P., dos Santos, C.T., Zamazal, O.: Results of the ontology alignment evaluation initiative 2013. In: Proceedings of the 8th Ontology Matching Workshop. pp. 61–100 (2013)
6. Jean-Mary, Y.R., Shironoshita, E.P., Kabuka, M.R.: Ontology matching with semantic verification. J. Web Sem. 7(3), 235–251 (2009)
7. Jiménez-Ruiz, E., Grau, B.C.: Logmap: Logic-based and scalable ontology matching. In: Proceedings of the 10th International Semantic Web Conference (ISWC). pp. 273–288 (2011)
8. Jiménez-Ruiz, E., Meilicke, C., Grau, B.C., Horrocks, I.: Evaluating mapping repair systems with large biomedical ontologies. In: Proceedings of the 26th International Workshop on Description Logics. pp. 246–257 (2013)
9. Meilicke, C.: Alignment incoherence in ontology matching. Ph.D. thesis, University Mannheim (2011)
10. Meilicke, C., Stuckenschmidt, H.: Incoherence as a basis for measuring the quality of ontology mappings. In: Proceedings of the 3rd Ontology Matching Workshop (OM) (2008)
11. Meilicke, C., Stuckenschmidt, H., Tamilin, A.: Repairing ontology mappings. In: Proceedings of the 22nd Conference on Artificial Intelligence (AAAI). pp. 1408–1413 (2007)
12. Motik, B., Shearer, R., Horrocks, I.: Hypertableau reasoning for description logics. Journal of Artificial Intelligence Research 36(1), 165–228 (2009)
13. Ngo, D., Bellahsene, Z., et al.: Yam++-a combination of graph matching and machine learning approach to ontology alignment task. Journal of Web Semantics 16 (2012)
14. Niepert, M., Noessner, J., Stuckenschmidt, H.: Log-linear description logics. In: Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI). pp. 2153–2158 (2011)
15. Noessner, J.: Efficient Maximum A-Posteriori Inference in Markov Logic and Application in Description Logics. Ph.D. thesis, University Mannheim (2014)
16. Noessner, J., Niepert, M.: Elog: A probabilistic reasoner for owl el. In: Proceedings of the 5th Conference on Web Reasoning and Rule Systems (RR). pp. 281–286 (2011)
17. Noy, N., Klein, M.: Ontology evolution: Not the same as schema evolution. Knowledge and Information System 6(4), 428440 (2004)
18. Noy, N.F., Musen, M.A.: Algorithm and tool for automated ontology merging and alignment. In: Proceedings of the 17th National Conference on Artificial Intelligence (2000)
19. Rahm, E., Bernstein, P.A.: A survey of approaches to automatic schema matching. VLDB Journal 10 (2001)
20. Reul, Q., Pan, J.Z.: Kosimap: Use of description logic reasoning to align heterogeneous ontologies. In: 23rd International Workshop on Description Logics DL2010. p. 489. Citeseer (2010)
21. Šváb, O., Svátek, V., Berka, P., Rak, D., Tomášek, P.: Ontofarm: Towards an experimental collection of parallel ontologies. Poster Track of ISWC (2005)

Time-Efficient Execution of Bounded Jaro-Winkler Distances

Kevin Dreßler and Axel-Cyrille Ngonga Ngomo

University of Leipzig
AKSW Research Group
Augustusplatz 10, 04103 Leipzig
Germany

Abstract. Over the last years, time-efficient approaches for the discovery of links between knowledge bases have been regarded as a key requirement towards implementing the idea of a Data Web. A considerable portion of the information contained available as RDF on the Web pertains to persons. Thus, efficient and effective measures for comparing names are central to facilitate the integration of information about persons on the Web of Data. The Jaro-Winkler measure has been developed especially for the purpose of comparing person names. Hence, we present a novel approach for the efficient comparison of sets of strings using this measure. We evaluate our approach on several datasets derived from DBpedia 3.9 and containing up to 10^5 strings and show that it scales linearly with the size of the data for large thresholds. We also evaluate our approach against SILK and show that we outperform it even on small datasets.

1 Introduction

The Linked Open Data Cloud (LOD Cloud) has developed to a compendium of more than 2000 datasets over the last few years.[1] Currently, data sets pertaining to more than 14 million persons have already been made available on the Linked Data Web.[2] While this number is impressive on its own, it is well known that the population of the planet has surpassed 7 billion people. Hence, the Web of Data contains information on less that 1% of the overall population of the planet (counting both the living and the dead). The output of open-government movements,[3] scientific conferences,[4] health data[5] and similar endeavours yet promises to make massive amounts of data pertaining to persons available in the near future. Dealing with this upcoming increase of the number of person-related resources requires providing means to integrate these datasets with the aim to facilitate statistical analysis, data mining, personlization, etc. However, while the

[1] See `http://stats.lod2.eu` for an overview of the current state of the Cloud. Last access: July 11th, 2014.
[2] Data collected from `http://stats.lod2.eu`. Last access: July 11th, 2014.
[3] See for example `http://data.gov.uk/`.
[4] See for example `http://data.semanticweb.org/`
[5] `http://aksw.org/Projects/GHO`

number of datasets on the Linked Data Web grows drastically, the number of links between datasets still stagnates.[6] Addressing this lack of links requires solving two main problems: the quadratic time complexity of link discovery (efficiency) and the automatic support of the detection of link specifications (effectiveness). In this paper, we address the efficiency of the execution of bounded Jaro-Winkler measures,[7] which are known to be effective when comparing person names [10]. To this end, we derive equations that allow discarding a large number of computations while executing bounded Jaro-Winkler comparisons with high thresholds.

The contributions of this paper are as follows:

1. We derive length- and range-based filters that allow reducing the number of strings t that are compared with a string s .
2. We present a character-based filter that allows detecting whether two strings s and t share enough resemblance to be similar according to the Jaro-Winkler measure.
3. We evaluate our approach w.r.t. to its runtime and its scalability with several threshold settings and dataset sizes.

The rest of this paper is structured as follows: In Section 2, we present the problem we tackled as well as the formal notation necessary to understand this work. In the subsequent Section 3, we present the three approaches we developed to reduce the runtime of bounded Jaro-Winkler computations. We then evaluate our approach in Section 4. Related work is presented in Section 5, where we focus on approaches that aim to improve the time-efficiency of link discovery. We conclude in Section 6. The approach presented herein is now an integral part of LIMES.[8]

2 Preliminaries

In the following, we present some of the symbols and terms used within this work.

2.1 Link Discovery

In this work, we use *link discovery* as a hypernym for deduplication, record linkage, entity resolution and similar terms used across literature. The formal specification of link discovery adopted herein is tantamount to the definition proposed in [16]: Given a set S of source resources, a set T of target resources and a relation R, our goal is to find the set $M \subseteq S \times T$ of pairs (s, t) such that $R(s, t)$. If R is `owl:sameAs`, then we are faced with a *deduplication task*. Given that the explicit computation of M is usually a very complex endeavour, M is most commonly approximated by a set $M' = \{(s, t, \delta(s, t)) \in S \times T \times \mathbb{R}^+ : \sigma(s, t) \geq \theta\}$, where σ is a (potentially complex) similarity function and $\theta \in [0, 1]$ is a similarity threshold. Given that this problem is in $O(n^2)$, using naïve algorithms to compare large S and T is most commonly impracticable. Thus, time-efficient approaches for the computation of bounded measures

[6] `http://linklion.org`

[7] We use bounded measures in the same sense as [13], i.e., to mean that we are only interested in pairs of strings whose similarity is greater than or equal to a given lower bound.

[8] `http://limes.sf.net`

have been developed over the last years for measures such as the Levenshtein distance, Minkowski distances, trigrams and many more [15].

In this paper, we thus study the following problem: Given a threshold $\theta \in [0, 1]$ and two sets of strings S and T, compute the set $M' = \{(s, t, \delta(s, t)) \in S \times T \times \mathbb{R}^+ : \sigma(s, t) \geq \theta\}$. Two categories of approaches can be considered to improve the runtime of measures: Lossy approaches return a subset M'' of M' which can be calculated efficiently but for which there are no guarantees that $M'' = M'$. Lossless approaches on the other hand ensure that their result set M'' is exactly the same as M'. In this paper, we present a lossless approach. To the best of our knowledge, only one other link discovery framework implements a lossless approach that has been designed to exploit the bound defined by the threshold θ to ensure a more efficient computation of the Jaro-Winkler distance, i.e., the SILK framework with the approach MultiBlock [9]. We thus compare our approach with SILK 2.6.0 in the evaluation section of this paper.

2.2 The Jaro-Winkler Similarity

Let Σ be the set of all the strings that can be generated by using an alphabet A. The Jaro measure $d_j : \Sigma \times \Sigma \to [0, 1]$ is a string similarity measure approach which was developed originally for name comparison in the U.S. Census. This measure takes into account the number of character matches m and the ratio of their transpositions t:

$$d_j = \begin{cases} 0 & \text{if } m = 0 \\ \frac{1}{3}\left(\frac{m}{|s_1|} + \frac{m}{|s_2|} + \frac{m-t}{m}\right) & \text{otherwise} \end{cases} \tag{1}$$

Here two characters are considered to be a match if and only only if (1) they are the same and (2) they are at most at a distance $w = \lfloor \frac{\max(|s_1|,|s_2|)}{2} \rfloor$ from each other. For example, for $s_1 = "Spears"$ and $s_2 = "Pears"$, the second s of s_1 matches the s of s_2 while the first s of s_1 does not match the s of s_2.

The Jaro-Winkler measure [27] is an extension of the Jaro distance. This extension is based on Winkler's observation that typing errors occur most commonly in the middle or at the end of a word, but very rarely in the beginning. Hence, it is legitimate to put more emphasis on matching prefixes if the Jaro distance exceeds a certain "boost threshold" b_t, originally set to 0.7.

$$d_w = \begin{cases} d_j & \text{if } d_j < b_t \\ d_j + (\ell p(1 - d_j)) & \text{otherwise} \end{cases} \tag{2}$$

Here, ℓ denotes the length of the common prefix and p is a weighting factor. Winkler uses $p = 0.1$ and $\ell \leq 4$. Note that ℓp must not be greater than 1.

For the strings $s_1 = "DEMOCRACY", s_2 = "DEMOGARPHY"$ (with s_2 being intentionally misspelled) we get the following output of the Jaro-Winkler measure.

- $|s_1| = 9, |s_2| = 10$
- $w = 4$
- $m = 7$

- $t = 1$
- $d_j = \frac{1}{3}\left(\frac{m}{|s_1|} + \frac{m}{|s_2|} + \frac{m-t}{m}\right) = \frac{1}{3}\left(\frac{7}{9} + \frac{7}{10} + \frac{6}{7}\right) = 0.778$
- $d_w = d_j + \ell p\,(1 - d_j) = 0.867$

3 Improving the Runtime of Bounded Jaro-Winkler

The main principle behind reducing the runtime of the computation of measures is to reduce their reduction ratio. Here, we use a sequence of filters that allow discarding similarity computations while being sure that they would have led to a similarity score which would have been less than our threshold θ. To this end, we regard the problem as that of finding filters that return an upper bound estimation $\theta_e(s_1, s_2) \geq d_w(s_1, s_2)$ for some properties of the input strings that can be computed in constant time. For a given threshold θ, if $\theta_e(s_1, s_2) \leq \theta$, then we can safely ignore the input (s_1, s_2).

3.1 Length-based filters

In the following, we denoted the length of a string s with $|s|$. Our first filter is based on the insight that large length differences are a guarantee for poor similarity. For example, the strings "a" and "$alpha$" cannot have a Jaro-Winkler similarity of 1 by virtue of their length difference. We can formalize this idea as follows: Let s_1 and s_2 be strings with respective lengths $|s_1|$ and $|s_2|$. Without loss of generality, we will assume that $|s_1| \leq |s_2|$. Moreover, let m be the number of matches across s_1 and s_2. Because $m \leq |s_1|$, we can substitute m with $|s_1|$ and gain the following upper bound estimation for $d_j(s_1, s_2)$:

$$d_j = \frac{1}{3}\left(\frac{m}{|s_1|} + \frac{m}{|s_2|} + \frac{m-t}{m}\right) \leq \frac{1}{3}\left(1 + \frac{|s_1|}{|s_2|} + \frac{|s_1| - t}{|s_1|}\right) \tag{3}$$

Now the lower bound for the number t of transpositions is 0. Thus, we obtain the following equation.

$$d_j \leq \frac{1}{3}\left(1 + \frac{|s_1|}{|s_2|} + 1\right) \leq \frac{2}{3} + \frac{|s1|}{3|s2|} \tag{4}$$

The application of this approximation on Winkler's extension is trivial:

$$d_w = d_j + \ell \cdot p \cdot (1 - d_j) \leq \frac{2}{3} + \frac{|s1|}{3|s2|} + \ell \cdot p \cdot \left(\frac{1}{3} - \frac{|s1|}{3|s2|}\right) = \theta_e \tag{5}$$

Consider the pair $s_1 =$ "$bike$" and $s_2 =$ "$bicycle$" and a threshold $\theta = 0.9$. Applying the estimation for Jaro we get $d_j \leq \frac{2}{3} - \frac{4}{3 \cdot 7} = 0.857$. This exceeeds the boost threshold, so we use equation 5 to compute $\theta_e(s_1, s_2) = 0.885$. Now we do not have to actually compute $d_w(s_1, s_2)$, since $\theta_e(s_1, s_2) < \theta$.

By using this approach we can decide in $O(1)$[9] if a given pairs score is greater than a given threshold, which saves us the much more expensive score computation for a big number of pairs, provided that the input strings sufficiently vary in length.

[9] In most programming languages, especially Java (which we used for our implementation), the length of string is stored in a variable and can thus be accessed in constant time.

3.2 Filtering ranges by length

The approach described above can be reversed to limit the number of pairs that we are going to be iterated over. To this end, we can construct a $index : \mathbb{N} \rightarrow 2^{\Sigma}$ which maps strings lengths $l \in \mathbb{N}$ to all strings s with $|s| = l$. With the help of this index, we can now determine the set of strings t that should be compared with the subset $S(l)$ of S that only contains strings of length l. We go about using this insight by computing the upper and lower bound for the length of a string t that should be compared with a string s. This is basically equivalent to asking what is the minimum length difference $||s|-|t||$ so that $\theta \geq \theta_e(s,t)$ is satisfied. We transpose equation 5 to the following for our lower bound:

$$|t| \geq \left\lfloor 3|s|\frac{\theta - \ell p}{1 - \ell p} - 2|s| \right\rfloor \tag{6}$$

Analogously, we can derive the following upper bound:

$$|t| \leq \left\lceil \frac{|s|}{3\frac{\theta-\ell p}{1-\ell p} - 2} \right\rceil \tag{7}$$

For example, consider a list of strings S with equally distributed, distinct string lengths $(4, 7, 11, 18)$. Using Equation 6 and Equation 7 we obtain Table 1. Taking into account the last column of the table, we will save a total of $\frac{3}{8}$ comparisons.

Table 1. Bounds for distinct string lengths ($\theta = 0.9$)

$\lvert t\rvert$	$\lvert s\rvert_{min}$	$\lvert s\rvert_{max}$	sizes in range
4	2	8	$(4, 7)$
7	3	14	$(4, 7, 11)$
11	5	22	$(7, 11, 18)$
18	9	36	$(11, 18)$

3.3 Filtering by character frequency

An even more fine-grained approach can be chosen to filter out computations. Let $e : \Sigma \times A \rightarrow \mathbb{N}$ be the function with returns the number of occurrences of a given character c in a string s. For the strings s_1 and$_2$, the number of maximum possible matches m_{max} can be expressed as

$$m_{max} = \sum\nolimits_{c \in s_1} min(e(s_1, c), e(s_2, c)) \geq m \tag{8}$$

Consequently, we can now substitute m for m_{max} in the Jaro distance computation:

$$d_j(s_1, s_2) = \frac{1}{3}\left(\frac{m_{max}}{|s_1|} + \frac{m_{max}}{|s_2|} + \frac{m_{max} - t}{m_{max}}\right) \leq \frac{1}{3}\left(\frac{m_{max}}{|s_1|} + \frac{m_{max}}{|s_2|} + 1\right) \tag{9}$$

We can thus derive that $d_j(s_1, s_2) \geq \theta$ iff

$$m_{max} \geq \frac{(3\theta - 1)|s_1||s_2|}{|s_1| + |s_2|}. \tag{10}$$

For instance, let $s_1 = "astronaut", s_2 = "astrochimp"$. The retrieval of m_{max} ist shown in Table 2.

Table 2. Calculation of m_{max}

c	$e(s_1, c)$	$e(s_2, c)$	$min(e(s_1, c), e(s_2, c))$	m_{max}
a	2	1	1	1
c	0	1	0	1
h	0	1	0	1
i	0	1	0	1
m	0	1	0	1
n	1	0	0	1
o	1	1	1	2
p	0	1	0	2
r	1	1	1	3
s	1	1	1	4
t	2	1	1	5
u	1	0	0	5

The question that remains to answer is how well do these filters perform on real person data. We answer this question empirically in the subsequent section.

4 Evaluation

The aim of our evaluation was to study how well our approach performs on real data. We chose DBpedia 3.9 as a source of data for our experiments as it contains data pertaining to 1.1 million persons and thus allows for both fine-grained evaluations and scalability evaluations. All experiments where deduplication experiments, i.e., $S = T$. We considered the list of all `rdfs:label` in DBpedia in our runtime evaluation and scalability experiments. We also computed the runtime of our approach on up to 10^5 labels for our scalability experiments. All experiments were performed on a 2.5 GHz Intel Core i5 machine with 16GB RAM running OS X 10.9.3.

4.1 Runtime Evaluation

In our first series of experiments, we evaluated the runtime of all filter combinations against the naïve approach on a small dataset containing 1000 labels from DBpedia. The results of our evaluation are shown in Figure 4.1. This evaluation suggests that all filters outperform the naïve approach. Moreover, the combination of all filtersl lead

to the best overall runtime in most cases. Interestingly, the character-based filter leads to a significant reduction of the number of comparisons (see Figure 2) by more than 2 orders of magnitude. However, the runtime improvement is not as substantial. This result seems to indicate that the lookup in the character indexes is very time-demanding. We will thus aim to improve our character indexing in future work. Overall, the results on this dataset already shows that we outperform the naïve approach by more than an order of magnitude when θ is high. The runtimes on a larger sample of size 10^4 show an even better improvement (see Figure 3). This suggests that the relative improvement of our approach improves with the size of the problem.

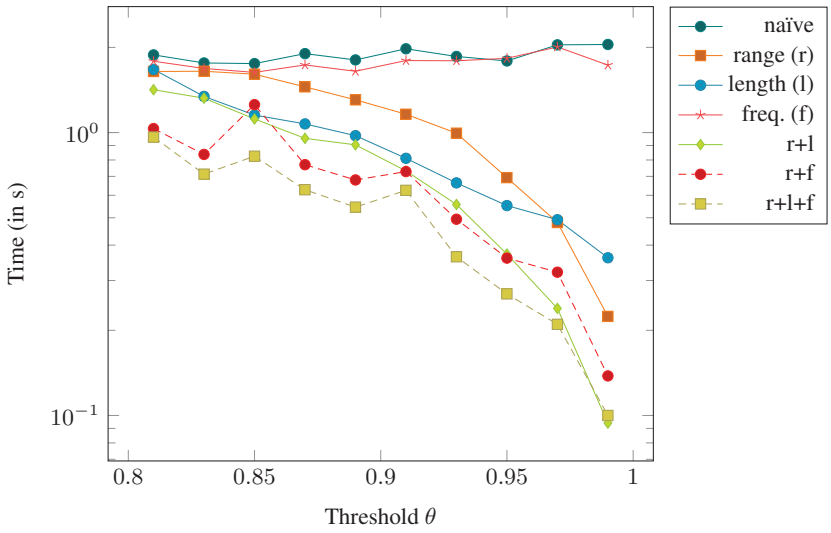


Fig. 1. Runtime comparison on input size 1000, scaling threshold

4.2 Scalability Evaluation

The aim of the scalability evaluation was to measure how well our approach deals with datasets of growing size datasets. In our first set of experiments, we looked at the growth of the runtime of our approach on datasets of growing sizes. Our results suggest that our approach grows linearly with the number of labels contained in S and T (see Figure 5). This suggests that the runtime of our approach can be easily predicted for large datasets, which of importance when asking users to wait for the results of the computation. The second series of scalability experiments looked at the runtime behaviour of our approach on a large dataset with 10^5 labels. Our results suggest that the runtime of our approach falls superlinearly with an increase of the threshold θ (see Figure 4). This behaviour suggest that our approach is especially useful on clean datasets, where high thresholds can be used for link discovery.

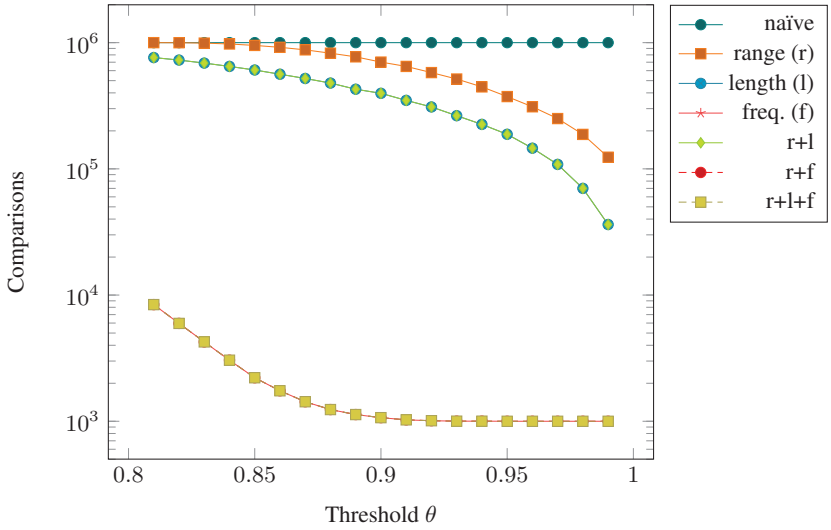


Fig. 2. Comparison of number of similarity computations on input size 1000, scaling threshold

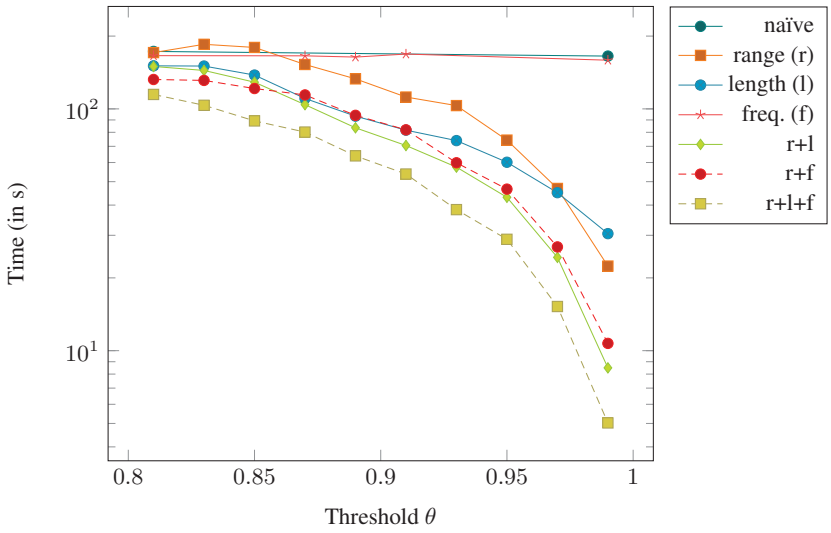


Fig. 3. Runtimes on sample of DBpedia labels with size 10^4, scaling threshold

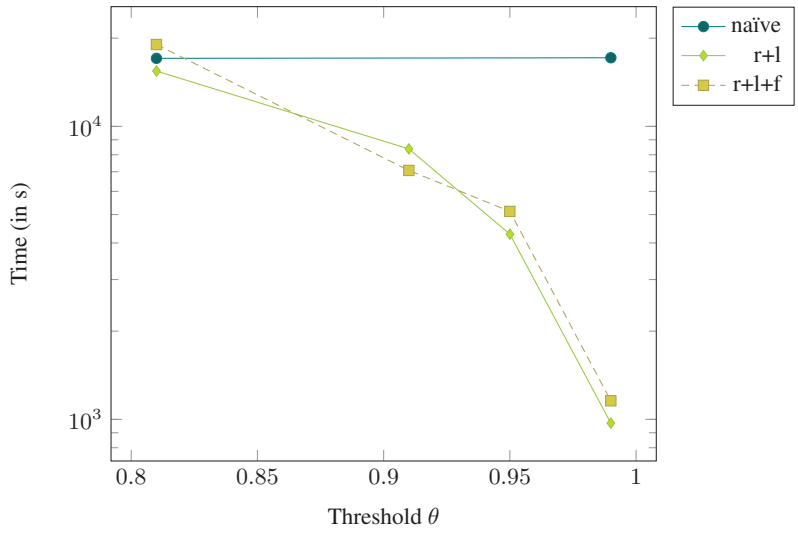


Fig. 4. Runtimes on sample of DBpedia labels with size 10^5, scaling threshold

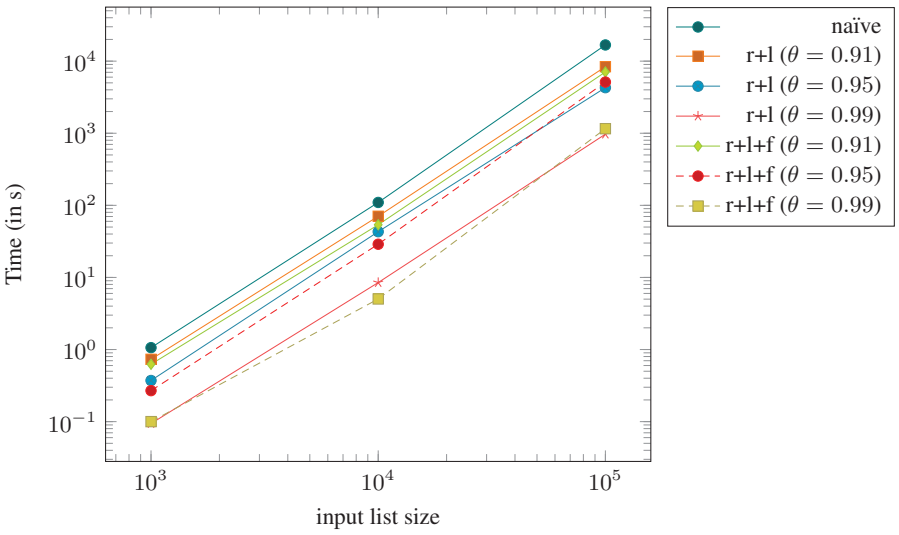


Fig. 5. Runtimes with multiple thresholds θ for growing input sizes

4.3 Comparison with existing approaches

We compared our approach with SILK2.6.0. To this end, we retrieved all `rdfs:label` of instances of subclasses of `Person`. We only compared with SILK on small datasets (i.e., on classes with small numbers of instances) as the results on these small datasets already showed that we outperform SILK consistently.[10] Our results are shown in Table 3. They suggest that the absolute difference in runtime grows with the size of the datasets. Thus, we did not consider testing larger datasets against SILK as in the best case, we were already 4.7 times faster than SILK (Architect dataset, $\theta = 0.95$).

Table 3. Runtimes (in seconds) of our approach (OA) and SILK 2.6.0

DBpedia Class	Size	OA(0.8)	OA(0.9)	OA(0.95)	SILK(0.8)	SILK(0.9)	SILK(0.95)
Actors	9509	15.07	10.13	6.38	27	25	25
Architect	3544	5.58	5.48	2.32	11	11	11
Criminal	5291	11.54	7.77	4.52	18	18	18

5 Related Work

The work presented herein is related to record linkage, deduplication, link discovery and the efficient computation of Hausdorff distances. An extensive amount of literature has been published by the database community on record linkage (see [11,6] for surveys). With regard to *time complexity*, time-efficient deduplication algorithms such as PPJoin+ [29], EDJoin [28], PassJoin [12] and TrieJoin [26] were developed over the last years. Several of these were then integrated into the hybrid link discovery framework LIMES [16]. Moreover, dedicated time-efficient approaches were developed for LD. For example, RDF-AI [24] implements a five-step approach that comprises the preprocessing, matching, fusion, interlink and post-processing of data sets. [17] presents an approach based on the Cauchy-Schwarz that allows discarding a large number of unnecessary computations. The approaches HYPPO [14] and $\mathcal{HR}^3$ [15] rely on space tiling in spaces with measures that can be split into independent measures across the dimensions of the problem at hand. Especially, $\mathcal{HR}^3$ was shown to be the first approach that can achieve a relative reduction ratio r' less or equal to any given relative reduction ratio $r > 1$. Standard blocking approaches were implemented in the first versions of SILK and later replaced with MultiBlock [9], a lossless multi-dimensional blocking technique. KnoFuss [20] also implements blocking techniques to achieve acceptable runtimes. Further approaches can be found in [25,4,21,22,7].

In addition to addressing the runtime of link discovery, several machine-learning approaches have been developed to learn link specifications (also called linkage rules) for link discovery. For example, machine-learning frameworks such as FEBRL [2] and MARLIN [1] rely on models such as Support Vector Machines [3] and decision

[10] We ran SILK with -Dthreads = 1 for the sake of fairness.

trees [23] to detect classifiers for record linkage. RAVEN [18] relies on active learning to detect linear or Boolean classifiers. The EAGLE approach [19] combines active learning and genetic programming to detect link specifications. KnoFuss [20] goes a step further and presents an unsupervised approach based on genetic programming for finding accurate link specifications. Other record deduplication approaches based on active learning and genetic programming are presented in [5,8].

6 Conclusion and Future Work

In this paper, we present a novel approach for the efficient execution of bounded Jaro-Winkler computations. Our approach is based on three filters which allow discarding a large number of comparisons. While our evaluation suggests that the filters are complementary, the character-based filter seems not to contribute to a significant reduction of the runtime once we deal with large datasets. We showed that our approach scales linearly with the amount of data it is faced with. Moreover, we showed that our approach can be make effective use of large thresholds by reducing the total runtime of the approach considerably. We also compared our approach with the state-of-the-art framework SILK 2.6.0 and showed that we outperform it on all datasets.
In future work, we will study the character-based filter in more detail and aim to eradicate its exact performace bottleneck. Moreover, we will evaluate partitioning of datasets and parallelization of filters to further improve the runtime of large datasets. Finally, we will test whether our approach improves the accuracy of specification detection algorithms such as EAGLE.

References

1. Mikhail Bilenko and Raymond J. Mooney. Adaptive duplicate detection using learnable string similarity measures. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '03, pages 39–48, New York, NY, USA, 2003. ACM.
2. Peter Christen. Febrl -: an open source data cleaning, deduplication and record linkage system with a graphical user interface. In *KDD*, pages 1065–1068, 2008.
3. Nello Cristianini and Elisa Ricci. Support vector machines. In *Encyclopedia of Algorithms*. 2008.
4. Philippe Cudré-Mauroux, Parisa Haghani, Michael Jost, Karl Aberer, and Hermann de Meer. idmesh: graph-based disambiguation of linked data. In *WWW*, pages 591–600, 2009.
5. J. De Freitas, G.L. Pappa, A.S. da Silva, M.A. Gonçalves, E. Moura, A. Veloso, A.H.F. Laender, and M.G. de Carvalho. Active learning genetic programming for record deduplication. In *Evolutionary Computation (CEC), 2010 IEEE Congress on*, pages 1–8. IEEE, 2010.
6. Ahmed K. Elmagarmid, Panagiotis G. Ipeirotis, and Vassilios S. Verykios. Duplicate record detection: A survey. *IEEE Trans. Knowl. Data Eng.*, 19(1):1–16, 2007.
7. Jérôme Euzenat, Alfio Ferrara, Willem Robert van Hage, et al. Results of the Ontology Alignment Evaluation Initiative 2011. In *OM*, 2011.
8. Robert Isele and Christian Bizer. Learning expressive linkage rules using genetic programming. *PVLDB*, 5(11):1638–1649, 2012.
9. Robert Isele, Anja Jentzsch, and Christian Bizer. Efficient Multidimensional Blocking for Link Discovery without losing Recall. In *WebDB*, 2011.

10. Jaro. Advances in record-linkage methodology as applied to matching the 1985 census of tampa, florida. Journal of the American Statistical Association 84(406):414–420, 1989.
11. Hanna Köpcke and Erhard Rahm. Frameworks for entity matching: A comparison. *Data Knowl. Eng.*, 69(2):197–210, 2010.
12. Guoliang Li, Dong Deng, Jiannan Wang, and Jianhua Feng. Pass-join: a partition-based method for similarity joins. *Proc. VLDB Endow.*, 5(3):253–264, November 2011.
13. Axel-Cyrille Ngonga Ngomo. Orchid - reduction-ratio-optimal computation of geo-spatial distances for link discovery. In *International Semantic Web Conference (1)*, pages 395–410, 2013.
14. Axel-Cyrille Ngonga Ngomo. A Time-Efficient Hybrid Approach to Link Discovery. In *OM*, 2011.
15. Axel-Cyrille Ngonga Ngomo. Link discovery with guaranteed reduction ratio in affine spaces with minkowski measures. In *International Semantic Web Conference (1)*, pages 378–393, 2012.
16. Axel-Cyrille Ngonga Ngomo. On link discovery using a hybrid approach. *J. Data Semantics*, 1(4):203–217, 2012.
17. Axel-Cyrille Ngonga Ngomo and Sören Auer. LIMES - A Time-Efficient Approach for Large-Scale Link Discovery on the Web of Data. In *IJCAI*, pages 2312–2317, 2011.
18. Axel-Cyrille Ngonga Ngomo, Jens Lehmann, Sören Auer, and Konrad Höffner. Raven - active learning of link specifications. In *OM*, 2011.
19. Axel-Cyrille Ngonga Ngomo and Klaus Lyko. Eagle: Efficient active learning of link specifications using genetic programming. In *ESWC*, pages 149–163, 2012.
20. Andriy Nikolov, Mathieu d'Aquin, and Enrico Motta. Unsupervised learning of link discovery configuration. In *ESWC*, pages 119–133, 2012.
21. Andriy Nikolov, Victoria Uren, Enrico Motta, and Anne Roeck. Overcoming schema heterogeneity between linked semantic repositories to improve coreference resolution. In *Proceedings of the 4th Asian Conference on The Semantic Web*, ASWC '09, pages 332–346, Berlin, Heidelberg, 2009. Springer-Verlag.
22. George Papadakis, Ekaterini Ioannou, Claudia Niederée, Themis Palpanas, and Wolfgang Nejdl. Eliminating the redundancy in blocking-based entity resolution methods. In *Proceedings of the 11th annual international ACM/IEEE joint conference on Digital libraries*, JCDL '11, pages 85–94, New York, NY, USA, 2011. ACM.
23. S. R. Safavian and D. Landgrebe. A survey of decision tree classifier methodology. *Systems, Man and Cybernetics, IEEE Transactions on*, 21(3):660–674, 1991.
24. Francois Scharffe, Yanbin Liu, and Chuguang Zhou. Rdf-ai: an architecture for rdf datasets matching, fusion and interlink. In *Proc. IJCAI 2009 workshop on Identity, reference, and knowledge representation (IR-KR), Pasadena (CA US)*, 2009.
25. Dezhao Song and Jeff Heflin. Automatically generating data linkages using a domain-independent candidate selection approach. In *International Semantic Web Conference (1)*, pages 649–664, 2011.
26. Jiannan Wang, Guoliang Li, and Jianhua Feng. Trie-join: Efficient trie-based string similarity joins with edit-distance constraints. *PVLDB*, 3(1):1219–1230, 2010.
27. William E. Winkler. String comparator metrics and enhanced decision rules in the fellegi-sunter model of record linkage. In *Proceedings of the Section on Survey Research*, pages 354–359, 1990.
28. Chuan Xiao, Wei Wang, and Xuemin Lin. Ed-Join: an efficient algorithm for similarity joins with edit distance constraints. *PVLDB*, 1(1):933–944, 2008.
29. Chuan Xiao, Wei Wang, Xuemin Lin, and Jeffrey Xu Yu. Efficient similarity joins for near duplicate detection. In *WWW*, pages 131–140, 2008.

A Two-step Blocking Scheme Learner for Scalable Link Discovery

Mayank Kejriwal and Daniel P. Miranker

University of Texas at Austin
{kejriwal,miranker}@cs.utexas.edu

Abstract. A two-step procedure for learning a link-discovery blocking scheme is presented. Link discovery is the problem of linking entities between two or more datasets. Identifying *owl:sameAs* links is an important, special case. A blocking scheme is a one-to-many mapping from entities to blocks. Blocking methods avoid $O(n^2)$ comparisons by clustering entities into blocks, and limiting the evaluation of link specifications to entity pairs within blocks. Current link-discovery blocking methods use blocking schemes tailored for *owl:sameAs* links or that rely on assumptions about the underlying link specifications. The presented framework learns blocking schemes for arbitrary link specifications. The first step of the algorithm is unsupervised and performs *dataset mapping* between a pair of dataset collections. The second supervised step *learns* blocking schemes on structurally heterogeneous dataset pairs. Application to RDF is accomplished by representing the RDF dataset in *property table* form. The method is empirically evaluated on four real-world test collections ranging over various domains and tasks.

Keywords: Heterogeneous Blocking, Instance Matching, Link Discovery

1 Introduction

With the advent of Linked Data, discovering links between entities has emerged as an active area of research [7]. Given a link specification, a naive approach would discover links by conducting $O(n^2)$ comparisons on the set of n entities. In the *Entity Resolution* (ER) community, a preprocessing technique called *blocking* mitigates full pairwise comparisons by clustering entities into blocks. Only entities within blocks are paired and compared [3]. Blocking is critical in data integration systems [5],[3].

Blocking methods require a *blocking scheme* to cluster entities. Advanced methods have been proposed to use a given blocking scheme effectively; relatively fewer works address the learning of blocking schemes. Even within ER, blocking scheme learners (BSLs) have met practical success only recently [2],[14],[10]. In the Semantic Web, the problem has received attention as scalably discovering *owl:sameAs* links [19]. ER is an important, but special, case of the *link discovery* problem, where the underlying link specification can be arbitrary. Such specifications can be learned, but brute-force applications would still be $O(n^2)$. Current

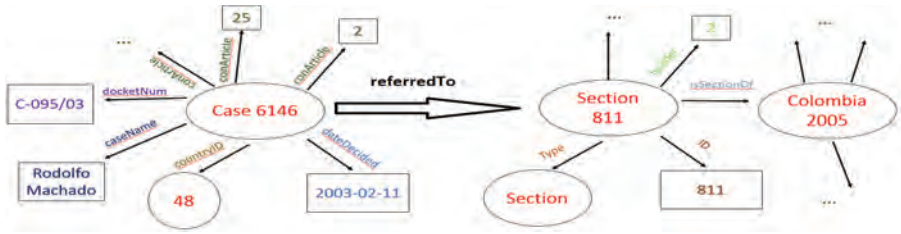


Fig. 1. Cases decided in Colombia must be *linked* to relevant sections of the constitution used in deciding that case. Only the single number *2* is relevant here for linking.

link discovery systems aim to be efficient by using token-based pre-clustering or metric space assumptions [7],[15].

Learning link-discovery blocking schemes for arbitrary underlying links remains unaddressed. Because the link can be arbitrary, a training corpus is required. Consider the example in Figure 1. Given a small number of such examples, the proposed BSL *adaptively* learns a scheme that covers true positives while reducing full quadratic cost, without relying on the formal link specification itself. Note that the learned blocking scheme is different from a learned link specification. In this paper, we exclusively address blocking.

In the Big Data era, *scalability*, *automation* and *heterogeneity* are essential components of systems and hence, practical requirements for real-world link discovery. Scalability is addressed by blocking, but current work assumes that the *dataset* pairs between which entities are to be linked are provided. In other words, datasets A and B are[1] input to the pipeline, and entities in A need to be linked to entities in B. Investigations in some important real-world domains show that pairs of dataset *collections* also need to undergo linking. Each collection is a *set* of datasets. An example is government data. Recent government efforts have led to release of public data as batches of files, as one of our real-world test sets demonstrates. Thus, *two* scalability issues are identified: at the collection level, and at the dataset level. That is, datasets in one collection first need to be mapped to datasets in the second collection, after which a blocking scheme is learned (and later, applied) on each mapped dataset pair.

Automation implies that human intervention needs to be kept to a minimum. In practice, this means that methods need to rely on fewer training examples. Finally, a heterogeneity issue arises if datasets in the collections are in two different data models, such as RDF and tabular.

The proposed BSL addresses these challenges in a combined setting. It takes as input two collections of datasets, with each collection an arbitrary mix of RDF and tabular datasets. The first step of the BSL performs unsupervised *dataset mapping* by relying on document similarity and efficient solutions to the Hungarian algorithm [9]. Each chosen dataset pair is input to the second step, which learns a link-discovery blocking scheme given a *constant* size training cor-

[1] If A and B are the same dataset, the problem is commonly denoted *deduplication*.

pus that does not require growing with the dataset. RDF datasets are reconciled with tabular datasets by representing them as *property tables* [22]. The problem thus reduces to learning schemes on tabular datasets with different schemas. Elmagarmid et al. refer to this as the *structural heterogeneity* problem [5]. To robustly deal with small, constant training sets, the BSL uses *bagging* [20]. To the best of our knowledge, this is the first paper that uses bagging, dataset mapping and property tables to address automation, scalability and heterogeneity respectively in the link-discovery blocking context.

The rest of this paper is structured as follows. Section 2 describes the related work in this area. Section 3 describes the property table representation and the BSL in detail. Section 4 describe the experimental results, and the paper concludes in Section 5.

2 Related Work

Link Discovery has been researched actively since the original Silk paper [21], which currently uses genetic programming to learn links [7]. Since we discuss learning of *blocking schemes*, most link specification learners are compatible, not competitive, with the proposed system. A full pipeline that can perform data fusion is RDF-AI [18]. We note that *active learning* techniques have been proposed to address the automation issue [16]; in this paper, we show a complementary solution using *bagging*.

Blocking has been extensively studied in the record linkage community [5], with a comprehensive survey by Christen [3]. Initial BSLs were supervised [2],[14]. Recently, an unsupervised feature selection based BSL was proposed by us, but assumed only *owl:sameAs* links and did not use bagging [10]. Token-based clustering has also been applied to the problem [13], together with Locality Sensitive Hashing (LSH) techniques [11]. However, LSH is usually applicable only to select distance measures like Jaccard or cosine. As such, it is more popularly applied to ontology matching [4],[6]. Other Semantic Web efforts for *owl:sameAs* include the approach by Song and Heflin [19]. Ma et al. proposed a system based on type semantics exclusively for ER on two individual datasets [12].

Multiple techniques for *using* blocking schemes have been investigated in the Semantic Web community [17]. An example (used in the Silk framework) is MultiBlock [8]. Another effort that assumes *metric* spaces is LIMES [15]. Finally, an advanced survey of bagging can be found in the work by Verikas et al. [20].

3 Algorithm

In this section, the two steps of the overall BSL in Figure 3 are described. Note that *dataset mapping* is an *optional*[2] step, but the second step (the core learner) is essential. As a preliminary, the property table representation is also summarized.

[2] Albeit empirically advantageous, when applied, as Section 4 will show.

Fig. 2. The property table representation. For subjects 'missing' a property, the reserved keyword *null* is entered. *;* is a reserved delimiter allowing fields to be *sets*.

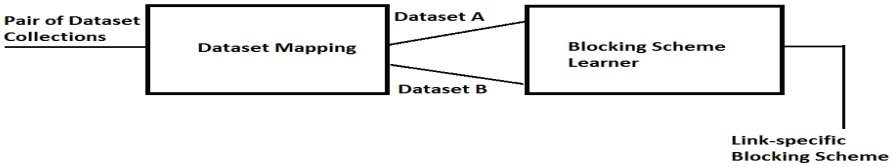


Fig. 3. The overall framework proposed in the paper.

3.1 Property Table Representation

Property tables were first proposed as *physical* data structures to efficiently implement triple stores [22]. This is the first application using them for link-discovery blocking. Figure 2 shows an example. Property tables reduce the problem of linking RDF and tabular datasets (and also RDF-RDF linkage) to tabular structural heterogeneity, that is, tables with different schemas. The rest of the paper assumes property table representation of RDF.

3.2 Dataset Mapping

The pseudocode for dataset mapping is given in Algorithm 1. The inputs to the algorithm are the dataset collections $\mathcal{R}$ and $\mathcal{S}$ and a boolean *confidence* function that is subsequently described. We define a dataset collection as a *set* of *independently released* datasets. Without loss of generality, assume that $|\mathcal{R}| \leq |\mathcal{S}|$. The output desired is a *confident* mapping $\mathcal{M} \subseteq \mathcal{R} \times \mathcal{S}$.

Algorithm 1 represents each dataset in each collection as a *term frequency* (TF) vector. The TF vector for each dataset is constructed by assigning a unique position to each distinct token and recording the count of that token in the dataset. Each TF vector is normalized by dividing each element by the total count of the respective token in the corresponding collection. A normalized TF vector is different from a TFIDF[3] vector.

[3] The TFIDF vector is constructed by dividing each element of a TF vector by the *number of datasets* in the corresponding collection in which the token occurred at least once (in lines 6 and 7) rather than the *total count of the token* in that collection.

A matrix $\boldsymbol{Q}$ is initialized with $\boldsymbol{Q}[i][j]$ containing the dot product of normalized TF vectors R_i and S_j. Once the matrix is constructed, the *max Hungarian* algorithm is invoked on the matrix, which has at least as many columns as rows, by the assumption above [9]. The algorithm must *assign* each row to some column, such that the sums of corresponding matrix entries are maximized. The problem is also equivalent to maximum weighted bipartite graph matching.

The confidence of each mapping is evaluated by $\mathcal{C}$. As an example of a confidence function, suppose the function returns *True* for a returned mapping (i, j) iff $\boldsymbol{Q}[i][j]$ is the *dominating* score, that is, greater than every score in its constituent row and column. Intuitively, this means that the mapping is not only the best possible, but also non-conflicting. In some sense, this assumes an aggressive strategy against false positives. Other[4] strategies can be formulated for other requirements; we leave these for future work.

Assuming the dominating strategy *a priori*, the Hungarian algorithm can be modified to terminate in linear time (in $\mathcal{R}$ and $\mathcal{S}$), otherwise it is cubic in the collection size [9]. Empirically, a reasonable confidence strategy would lead to savings if the total number of records is far greater than the number of datasets. For large collections, preferrable strategies should have theoretical guarantees, like the dominating strategy. We observed dataset mapping to achieve near-instantaneous runtime, even with a standard Hungarian implementation.

Intuitively, dataset mapping is expected to be a well-performing heuristic because constituent datasets are *independently* released. Algorithms like LIMES, Canopy Clustering and unsupervised methods benefit because they can cluster entities in isolated dataset pairs, rather than all the entities in the collection. With correct mapping, both quality and scalability are expected to improve. Experimentally, the gains are demonstrated in Section 4.

3.3 Heterogeneous Blocking Scheme Learner

Given two structurally heterogeneous tabular sources, the goal of the second step is supervised learning of a *heterogeneous* blocking scheme. In earlier works, *DNF blocking schemes* were found to be fairly representative and learned using set-covering algorithms [2],[14]. In recent work, we showed that a *feature selection* technique outperformed state-of-the-art DNF BSLs [10]. To summarize the work briefly, given training sets D and N containing duplicate and non-duplicate tuple pairs respectively, each pair is first converted into a vector with $O(m_1 m_2)$ *binary* features, where m_1, m_2 is the number of attributes in datasets R_1,R_2 respectively. Thus, two sets F_D and F_N containing labeled feature vectors are obtained. A set of features must now be chosen such that a minimum fraction ϵ of positives are covered, and with no individual feature covering more than a fraction η of negatives. For further details on these parameters and the feature conversion and selection process, we refer the reader to the original work [10].

Note that the originally proposed algorithm had no concept of bagging and assumed the training set was representative enough. Algorithm 2 shows how bag-

[4] For example, using a threshold to map multiple datasets to each other.

Algorithm 1 Perform dataset mapping.

Input: Dataset Collections $\mathcal{R}$ and $\mathcal{S}$, boolean confidence function $\mathcal{C}$
Output: A mapping $\mathcal{M}$ between $\mathcal{R}$ and $\mathcal{S}$

1. **for all** datasets $R_i \in \mathcal{R}$ **do**
   $\overrightarrow{T_i^{\mathcal{R}}}$:= Term Frequency vector of terms in R_i
2. **end for**
3. **for all** datasets $S_i \in \mathcal{S}$ **do**
   $\overrightarrow{T_i^{\mathcal{S}}}$:= Term Frequency vector of terms in S_i
4. **end for**
5. Construct vectors $\overrightarrow{T^{\mathcal{R},\mathcal{S}}}$, with j^{th} element $\overrightarrow{T^{\mathcal{R},\mathcal{S}}}[j] := \Sigma_i \overrightarrow{T_i^{\mathcal{R},\mathcal{S}}}[j]$
6. Normalize each $\overrightarrow{T_i^{\mathcal{R}}}$ by applying once, $\forall j, \overrightarrow{T_i^{\mathcal{R}}}[j] := \overrightarrow{T_i^{\mathcal{R}}}[j] / \overrightarrow{T^{\mathcal{R}}}[j]$
7. Normalize each $\overrightarrow{T_i^{\mathcal{S}}}$ by applying once, $\forall j, \overrightarrow{T_i^{\mathcal{S}}}[j] := \overrightarrow{T_i^{\mathcal{S}}}[j] / \overrightarrow{T^{\mathcal{S}}}[j]$
8. Initialize empty matrix $\boldsymbol{Q}$ with $|\mathcal{R}|$ rows and $|\mathcal{S}|$ columns
9. **for all** $i \in 1 \ldots |\mathcal{R}|$ **do**
   for all $j \in 1 \ldots |\mathcal{S}|$ **do**
   $\boldsymbol{Q}[i][j] := \overrightarrow{T_i^{\mathcal{R}}}.\overrightarrow{T_j^{\mathcal{S}}}$
   end for
10. **end for**
11. Let $\mathcal{M}$ be the results of running *max* Hungarian algorithm on $\boldsymbol{Q}$
12. **for all** $(i, j) \in \mathcal{M}$ **do**
   if applying $\mathcal{C}$ on $score(R_i, S_j)$ yields *False* **then**
   Remove (R_i, S_j) from $\mathcal{M}$
   end if
13. **end for**
14. **return** $\mathcal{M}$

ging can be incorporated, to work with small[5] training sets. Bagging parameters τ, β are now also input. β specifies the number of bagging iterations and τ is the sampling rate for bagging. In each bagging iteration, a fraction τ of the overall training sample is chosen to undergo feature selection by calling *FisherDisjunctive* (Algorithm 3 in [10]). A feature is technically a *specific blocking predicate* (SBP) e.g. *CommonToken(Last Name, Name)*. Intuitively, the SBP implies that two entities (from different datasets) with a common token in their respective *Last Name* and *Name* field values *share* a block. Features (or SBPs) chosen in each bagging iteration are added to $\mathcal{B}'$. The final DNF blocking scheme $\mathcal{B}$ is *heterogeneous* precisely because it accommodates two *different* schemas, as the SBP example above shows.

In some cases, training examples might not be available at all. If the task is learning *owl:sameAs* links, the automatic training set generator in our original work can be used to generate noisy training samples [10]. The rest of the procedure remains the same. We evaluate this scenario in one of our test suites.

[5] Small training sets affects learning algorithm quality, but compensate well for $O(m_1 m_2)$ feature-vector dimensionality. Bagging allows controlled compromise between scalability and quality.

Algorithm 2 Learn Link-Specific Blocking Scheme

Input : Positive feature-vectors set F_D, negative feature-vectors set F_N, coverage parameter ϵ, pruning parameter η, bagging iterations β, sampling size τ
Output : Blocking scheme $\mathcal{B}$

1. Initialize $\mathcal{B}' := \phi$
2. **for all** $iter = 1 \ldots \beta$ **do**
   Randomly sample (with replacement) $\tau|F_D|$ and $\tau|F_N|$ vectors from F_D and F_N, insert into new sets F'_N and F'_D respectively
   $\mathcal{B}' := \mathcal{B}' \cup FisherDisjunctive(F'_D, F'_N, \epsilon, \eta)$
3. **end for**
4. Output disjunction of elements in $\mathcal{B}'$ as $\mathcal{B}$

Table 1. Test dataset details. The notation, where applicable, is (first collection)/(second collection) or (first collection)×(second collection)

Collection	Number of datasets	Task	Total entity pairs	True positive pairs	Data model
Case Law/Constitute (Colombia)	1/2	non-ER	1204 × 2220 ≈ 2.67 million	5577	RDF/RDF
Case Law/Constitute (Venezuela)	1/2	non-ER	1503 × 1601 ≈ 2.4 million	555	RDF/RDF
JCT/Treasury	5/5	non-ER	1135 × 845 ≈ 1 million	24,227	Tab./Tab.
Dbpedia/vgchartz	1/1	ER	16740 × 20000 = 334.8 million	10,000	RDF/Tab.

4 Experiments

In this section, the algorithm is experimentally evaluated. Datasets, metrics and baseline are first described, followed by a set of results and a discussion.

4.1 Datasets

The algorithm is evaluated on four real-world dataset collections over three different domains, described in Table 1. In the first test set, the first collection consists of a single RDF dataset describing court cases decided in Colombia, along with various properties of those cases. The second collection has two RDF datasets, only one of which is relevant for linkage. This dataset describes article numbers (as literal object values) in the Colombia constitution[6]. The task is to predict links between the cases in the first collection and the articles in the second collection used to decide the case. An example was shown in Figure 1. The second test set is similar, but for Venezuela. The third test set consists of ten US

[6] `constituteproject.org`

government estimated budget datasets from 2009 to 2013, released separately by the Treasury department and the Joint Committee on Taxation. All datasets in this collection were published in tabular form, but the two collections are structurally heterogeneous. The goal is to link entities (describing a particular budget allocation) that share the same budget function (such as *health*) in the *same* year. These three test cases are proper dataset collections, given at least one collection contains more than one dataset. Other such collections can also be observed on the respective website[7].

The fourth test set contains collections derived from the *video games* domain and differs from the other three test sets in three important respects. First, the dataset mapping step is not applicable, since each collection only contains one dataset. Note that the datasets in this test set are large compared with the other test sets. Second, the first dataset is RDF and was queried from DBpedia[8] while the second dataset is tabular and from a popular charting website[9]. Finally, the noisy training set generator can be used, given the link is *owl:sameAs*. We have collected all publicly available test cases on a single portal[10], along with other implementation details such as the *features* (or SBPs) used in the experiments.

4.2 Metrics

We adopted two metrics, Pairs Completeness (PC) and Reduction Ratio (RR), from the blocking literature [3]. PC measures *recall* or *effectiveness* in the blocking setting; specifically, the ratio of true positives that have fallen within the block and the total number of true positives. RR is the percentage of comparisons that have been avoided compared to full quadratic cost and represents *efficiency*. For example, an RR of 99 percent means that the blocking scheme has reduced the complexity of the full pairwise task by that amount. Note that the *optimal* RR for 100 percent PC for the datasets in Table 1 can be calculated by using the formula $1 - C_5/C_4$ where C_p stands for Column p. Following previous research, PC-RR graphs are used to quantify the effectiveness-efficiency tradeoff [2].

4.3 Baseline

As earlier stated, many popular link discovery systems like Silk [7] use token-based pre-matching to reduce complexity. *Canopy Clustering*, originally proposed by McCallum et al. [13], represents such methods since it is token-based, makes few assumptions and has been shown to be experimentally robust [1]. It was also used as the baseline in another competitive system [2]; hence, we use it as our baseline. In the best performing implementation[11], the algorithm

[7] e.g. `http://www.pewstates.org/research/reports/` for the third test case.
[8] `dbpedia.org`
[9] `vgchartz.com`
[10] `https://sites.google.com/a/utexas.edu/mayank-kejriwal/datasets`
[11] Documented by Baxter et al. [1].

randomly chooses a seed entity from one dataset to represent a cluster, and all entities in the second dataset with TFIDF scores above a threshold are placed in that cluster. Clusters may overlap.

4.4 Methodology

The dataset mapping step was applied on the first three test sets in Table 1. It is not applicable to the fourth test set. The dominating strategy introduced earlier was used as the confidence function in Algorithm 1. For the second step, note that the baseline chooses seeds randomly. We compensated by conducting ten trials for each run of the algorithm, and averaging PC and RR. The threshold was tuned and set to a low value of 0.0005 to maximize baseline recall.

The parameters in Algorithm 2 were set to values that were found after some initial tuning (on a subset of the first test collection). The size of initial training set was set to 300 each for both duplicates ($|F_D|$) and non-duplicates ($|F_N|$). β was set to 10, τ to 30, ϵ to 0.8 and η to 0.2. In additional experiments, we varied each of these parameters by 50%. There was no significant difference from the results we subsequently show with these parameter settings. Future work will investigate automatic parameter tuning. In our earlier work, the feature selection was also found robust to varying parameters [10]. Note that the training set is kept constant, regardless of dataset growth.

To evaluate the learned blocking scheme in a practical blocking method, one additional parameter, *maxBucketPairs* is used to enable a technique called *block purging* [17]. The technique discards blocks that have more than *maxBucketPairs* candidate pairs, since the *cost* of processing these blocks is greater than *expected gain*. Block purging was also used in the baseline, for consistency. Varying *maxBucketPairs* from values typically ranging from 1000 to 100,000 effectively varies RR. Data points showing PC at different values of RR are obtained and plotted. All experiments were run on an Intel Core 2 Duo machine with 3 GB of memory and 2.4 GHz clock speed. All code was implemented in Java.

4.5 Results and Discussion

With the dominating strategy, dataset mapping yielded perfect mappings for all three test sets. We ran some additional experiments, including using more government test data (from years 2003-2013 instead of 2009-2013) and Constitute and Case Law data from other countries, and the mappings were still perfect. It would seem, therefore, that the normalized TF measures and the dominating strategy are suited to the problem, at least on the tested domains.

Figure 4 shows the PC-RR tradeoff results of the learned blocking scheme for Canopy Clustering and the proposed method both with and without dataset mapping[12], on the Colombia and Venezuela collections. The gains of dataset

[12] Recall that dataset mapping was designed to be compatible with other algorithms as well, including Canopy Clustering.

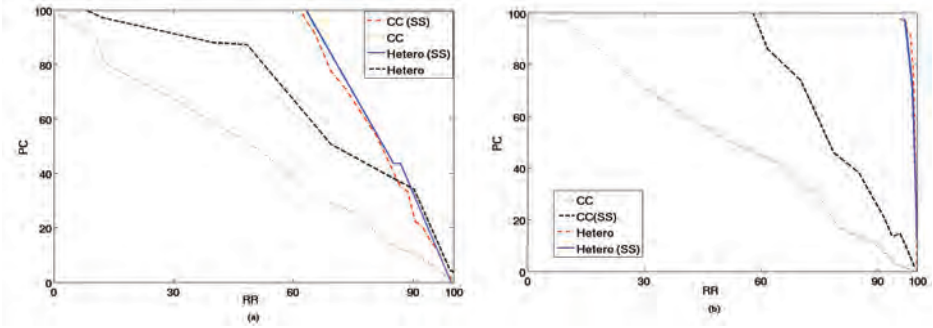


Fig. 4. Results of the proposed method (*Hetero*) against baseline (*CC*) on the (a) Colombia and (b) Venezuela datasets, with *SS* indicating that dataset mapping (or *Source Selection* as it is denoted in the codebase) was utilized. *PC* is Pairs Completeness and *RR* is Reduction Ratio.

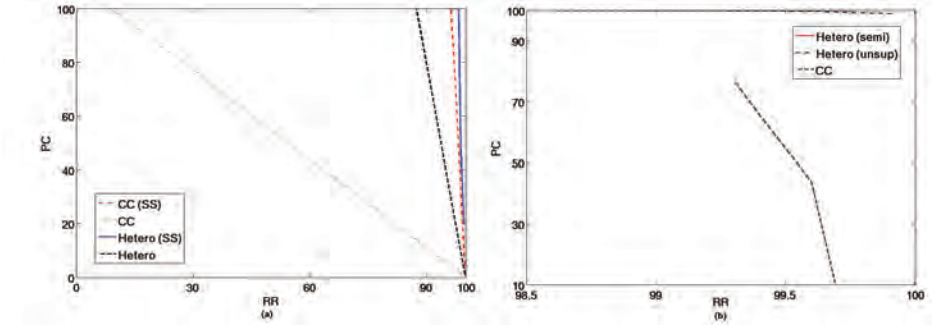


Fig. 5. *Hetero* vs. *CC* on the (a) government 2009-13 budget and (b) video game datasets. In (b), the underlying link was *owl:sameAs*. The noisy training set generator was used for the *unsup* version of *Hetero*, while perfectly labeled examples were provided for *semi*. Note the changed scale (esp. X-axis) in (b).

mapping are readily apparent for *CC* in both cases. The proposed method, *Hetero*, outperforms the non dataset-mapping version of *CC* but the gap narrows considerably when dataset mapping is employed. This shows that, in cases where training sets are not readily available, an off-the-shelf dataset mapping algorithm can boost performance. The dataset mapping gains for *Hetero* aren't significant on Venezuela, mainly because the algorithm performs well on this dataset even without mapping. On *CC*, however, the gains are again apparent. Note that Venezuela (*(b)* in Figure 4) represents some of the challenges of doing link discovery versus just ER. The proposed BSL was able to overcome these challenges by employing bagging and feature selection.

Figure 5a shows the results on the five-pair government budget data. For this collection, the gains of dataset mapping are amplified. This is because there are more datasets in the collection, so the dataset mappings are particularly useful.

We ran further experiments on the full government data (from years 2003-2013) and confirmed this. This time, *Hetero* also shows noticeable gains, with the curve shifting to the right when dataset mapping is employed. Figure 5b shows the results for learning blocking schemes for ER. The BSL is able to significantly outperform *CC*, regardless of whether it is completely unsupervised or with a provided perfectly labeled training set.

Finally, we repeated the experiments above but *without* bagging. Highest *f-scores*[13] of PC and RR declined on all cases by at least 5%, with 95% statistical significance using Student's distribution. Otherwise, the graphical trends were similar. We do not repeat the figures here.

5 Future Work and Conclusion

In this paper, a link-discovery blocking scheme learner was proposed. The first step of the method operates in an unsupervised fashion and performs *dataset mapping* by employing document-level similarity measures. It is compatible with existing clustering and blocking algorithms, experimental savings demonstrated on two such methods. The second step is a heterogeneous BSL that uses techniques like bagging to achieve robust performance, even as the training sets remain constant and the datasets grow in size.

Future work will evaluate the dataset mapping step and the accompanying confidence strategies more extensively, and develop parameter tuning techniques for the learner itself. Another important aspect is investigating scalability of the learner; in particular, we are developing techniques for 'pruning' property tables so that the learner can efficiently scale by learning schemes in a reduced feature space. We believe that this provides an excellent opportunity for cross-fertilizing ongoing scalability efforts in the ontology matching community [6].

Acknowledgments. The authors would like to thank Juan Sequeda for providing the Constitute and Case Law datasets.

References

1. R. Baxter, P. Christen, and T. Churches. A comparison of fast blocking methods for record linkage. In *ACM SIGKDD*, volume 3, pages 25–27. Citeseer, 2003.
2. M. Bilenko, B. Kamath, and R. J. Mooney. Adaptive blocking: Learning to scale up record linkage. In *Data Mining, 2006. ICDM'06. Sixth International Conference on*, pages 87–96. IEEE, 2006.
3. P. Christen. A survey of indexing techniques for scalable record linkage and deduplication. *Knowledge and Data Engineering, IEEE Transactions on*, 24(9):1537–1555, 2012.
4. 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 *The Semantic Web–ISWC 2012*, pages 49–64. Springer, 2012.

[13] Given by 2.RR.PC/(PC+RR)

5. A. K. Elmagarmid, P. G. Ipeirotis, and V. S. Verykios. Duplicate record detection: A survey. *Knowledge and Data Engineering, IEEE Transactions on*, 19(1):1–16, 2007.
6. J. Euzenat, P. Shvaiko, et al. *Ontology matching*, volume 18. Springer, 2007.
7. R. Isele and C. Bizer. Learning expressive linkage rules using genetic programming. *Proceedings of the VLDB Endowment*, 5(11):1638–1649, 2012.
8. R. Isele, A. Jentzsch, and C. Bizer. Efficient multidimensional blocking for link discovery without losing recall. In *WebDB*, 2011.
9. R. Jonker and T. Volgenant. Improving the hungarian assignment algorithm. *Operations Research Letters*, 5(4):171–175, 1986.
10. M. Kejriwal and D. P. Miranker. An unsupervised algorithm for learning blocking schemes. In *Data Mining, 2013. ICDM'13. Thirteenth International Conference on*. IEEE, 2013.
11. H.-s. Kim and D. Lee. Harra: fast iterative hashed record linkage for large-scale data collections. In *Proceedings of the 13th International Conference on Extending Database Technology*, pages 525–536. ACM, 2010.
12. Y. Ma, T. Tran, and V. Bicer. Typifier: Inferring the type semantics of structured data. In *Data Engineering (ICDE), 2013 IEEE 29th International Conference on*, pages 206–217. IEEE, 2013.
13. A. McCallum, K. Nigam, and L. H. Ungar. Efficient clustering of high-dimensional data sets with application to reference matching. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 169–178. ACM, 2000.
14. M. Michelson and C. A. Knoblock. Learning blocking schemes for record linkage. In *Proceedings of the National Conference on Artificial Intelligence*, volume 21, page 440. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2006.
15. A.-C. N. Ngomo. A time-efficient hybrid approach to link discovery. *Ontology Matching*, page 1, 2011.
16. A.-C. N. Ngomo and K. Lyko. Eagle: Efficient active learning of link specifications using genetic programming. In *The Semantic Web: Research and Applications*, pages 149–163. Springer, 2012.
17. G. Papadakis, E. Ioannou, C. Niederée, and P. Fankhauser. Efficient entity resolution for large heterogeneous information spaces. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 535–544. ACM, 2011.
18. F. Scharffe, Y. Liu, and C. Zhou. Rdf-ai: an architecture for rdf datasets matching, fusion and interlink. In *Proc. IJCAI 2009 workshop on Identity, reference, and knowledge representation (IR-KR), Pasadena (CA US)*, 2009.
19. D. Song and J. Heflin. Automatically generating data linkages using a domain-independent candidate selection approach. In *The Semantic Web–ISWC 2011*, pages 649–664. Springer, 2011.
20. A. Verikas, A. Gelzinis, and M. Bacauskiene. Mining data with random forests: A survey and results of new tests. *Pattern Recognition*, 44(2):330–349, 2011.
21. J. Volz, C. Bizer, M. Gaedke, and G. Kobilarov. Discovering and maintaining links on the web of data. In *The Semantic Web-ISWC 2009*, pages 650–665. Springer, 2009.
22. K. Wilkinson, C. Sayers, H. A. Kuno, D. Reynolds, et al. Efficient rdf storage and retrieval in jena2. In *SWDB*, volume 3, pages 131–150, 2003.

Results of the Ontology Alignment Evaluation Initiative 2014*

Zlatan Dragisic[2], Kai Eckert[3], Jérôme Euzenat[4], Daniel Faria[12], Alfio Ferrara[5], Roger Granada[6,7], Valentina Ivanova[2], Ernesto Jiménez-Ruiz[1], Andreas Oskar Kempf[8], Patrick Lambrix[2], Stefano Montanelli[5], Heiko Paulheim[3], Dominique Ritze[3], Pavel Shvaiko[9], Alessandro Solimando[11], Cássia Trojahn[7], Ondřej Zamazal[10], and Bernardo Cuenca Grau[1]

[1] University of Oxford, UK
{berg,ernesto}@cs.ox.ac.uk
[2] Linköping University & Swedish e-Science Research Center, Linköping, Sweden
{zlatan.dragisic,valentina.ivanova,patrick.lambrix}@liu.se
[3] University of Mannheim, Mannheim, Germany
{kai,heiko,dominique}@informatik.uni-mannheim.de
[4] INRIA & Univ. Grenoble-Alpes, Grenoble, France
Jerome.Euzenat@inria.fr
[5] Università degli studi di Milano, Italy
{alfio.ferrara,stefano.montanelli}@unimi.it
[6] Pontifícia Universidade Católica do Rio Grande do Sul, Porto Alegre, Brazil
roger.granada@acad.pucrs.br
[7] IRIT & Université Toulouse II, Toulouse, France
cassia.trojahn@irit.fr
[8] GESIS – Leibniz Institute for the Social Sciences, Cologne, Germany
andreas.kempf@gesis.org
[9] TasLab, Informatica Trentina, Trento, Italy
pavel.shvaiko@infotn.it
[10] University of Economics, Prague, Czech Republic
ondrej.zamazal@vse.cz
[11] DIBRIS, University of Genova, Italy
alessandro.solimando@unige.it
[12] LASIGE, Faculdade de Ciências, Universidade de Lisboa, Portugal
dfaria@xldb.di.fc.ul.pt

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 2014 offered 7 tracks with 9 test cases followed by 14 participants. Since 2010, the campaign has been using a new evaluation modality which provides more automation to the evaluation. This paper is an overall presentation of the OAEI 2014 campaign.

* This paper improves on the "Preliminary results" initially published in the on-site proceedings of the ISWC workshop on Ontology Matching (OM-2014). The only official results of the campaign, however, are on the OAEI web site.

1 Introduction

The Ontology Alignment Evaluation Initiative[1] (OAEI) is a coordinated international initiative, which organizes the evaluation of the increasing number of ontology matching systems [12, 15]. 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) [34]. 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 2013 the OAEI campaigns were held at the Ontology Matching workshops collocated with ISWC [13, 11, 4, 8–10, 1, 6]. In 2014, the OAEI results were presented again at the Ontology Matching workshop[2] collocated with ISWC, in Riva del Garda, Italy.

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 2014, almost all of the OAEI data sets were evaluated under the SEALS modality, providing a more uniform evaluation setting.

This paper synthetizes the 2014 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 3-10 discuss the settings and the results of each of the test cases. Section 12 overviews lessons learned from the campaign. Finally, Section 13 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).

2.1 Tracks and test cases

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

[1] http://oaei.ontologymatching.org
[2] http://om2014.ontologymatching.org
[3] http://www.seals-project.eu

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 two new benchmarks based on different ontologies.

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

Anatomy (§4): The anatomy real world 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.

Directories and thesauri

Library (§8): The library test case is a real-world task to match two thesauri. The goal of this test case is to find whether the matchers can handle such lightweight ontologies including a huge amount of concepts and additional descriptions. Results are evaluated both against a reference alignment and through manual scrutiny.

Interactive matching

Interactive (§9): This test case offers the possibility to compare different interactive matching tools which require 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. All participating systems are evaluated on the conference data set using an oracle based on the reference alignment.

Ontology Alignment For Query Answering OA4QA (§10): 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.

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	√
large bio	OWL	=	[0 1]	open	EN	√
multifarm	OWL	=	[0 1]	open	CZ, CN, DE, EN, ES, FR, NL, RU, PT	√
library	OWL	=	[0 1]	open	EN, DE	√
interactive	OWL	=, <=	[0 1]	open	EN	√
OA4QA	OWL	=, <=	[0 1]	open	EN	
im-identity	OWL	=	[0 1]	blind	EN, IT	√
im-similarity	OWL	<=	[0 1]	blind	EN, IT	√

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).

Instance matching

Identity (§11): The identity task is a typical evaluation task of instance matching tools where the goal is to determine when two OWL instances describe the same real-world entity.

Similarity(§11): The similarity task focuses on the evaluation of the similarity degree between two OWL instances, even when they describe different real-world entities. Similarity recognition is new in the instance matching track of OAEI, but this kind of task is becoming a common issue in modern web applications where large quantities of data are daily published and usually need to be classified for effective fruition by the final user.

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 had to be uploaded on the SEALS portal[4]. Consequently, the evaluation was 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}, 2014. This gave

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

potential participants the occasion to send observations, bug corrections, remarks and other test cases to the organizers. The goal of this preparatory period is to ensure that the delivered tests make sense to the participants. The final test base was released on July 3^{rd}, 2014. 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 [7]. 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}, 2014.

2.5 Evaluation phase

Participants have been encouraged to upload their wrapped tools on the SEALS portal by September 1^{st}, 2014. 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}, 2014. 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.

2.6 Comments on the execution

The number of participating systems has regularly increased over the years: 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. However, 2014 has suffered a significant decrease with only 14 systems. However, participating systems are now constantly changing. In 2013, 11 (7 in 2012) 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).

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.

System	AML	AOT	AOTL	InsMT	InsMTL	LogMap	LogMap-Bio	LogMapLt	LogMap-C	MaasMatch	OMReasoner	RiMOM-IM	RSDLWB	XMap	Total=14
Confidence	√	√	√			√	√		√	√		√			8
benchmarks	√	√	√			√		√	√	√	√		√	√	10
anatomy	√	√	√			√	√	√	√	√			√	√	10
conference	√	√	√			√		√	√	√	√		√	√	10
multifarm	√					√								√	3
library	√					√		√	√	√			√	√	7
interactive	√					√									2
large bio	√	√	√			√	√	√	√	√	√		√	√	11
OA4QA	√	√	√			√		√	√	√	√		√	√	10
instance				√	√	√			√			√			5
total	8	5	5	1	1	9	2	6	7	6	4	1	6	7	68

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.

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 [14] from different seed ontologies. This year, we used three ontologies:

biblio The bibliography ontology used in the previous years which concerns bibliographic references and is inspired freely from BibTeX;
cose COSE[5] is the Casas Ontology for Smart Environments;
dog DogOnto[6] is an ontology describing aspects of intelligent domotic environments.

The characteristics of these ontologies are described in Table 3.

The test cases were not available to participants. They still could test their systems with respect to previous year data sets, but they have been evaluated against newly

[5] http://casas.wsu.edu/owl/cose.owl
[6] http://elite.polito.it/ontologies/dogont.owl

Test set	biblio	cose	dog
classes+prop	33+64	196	842
instances	112	34	0
entities	209	235	848
triples	1332	690	10625

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

generated tests. The tests were also blind for the organizers since we did not looked 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

Evaluations 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 1.7 and a maximum heap size of 8GB (which has been necessary for the larger tests, i.e., dog). No timeout was explicitly set.

Reported figures are the average of 5 runs. As has already been shown in [14], there is not much variance in compliance measures across runs. This is not necessarily the case for time measurements so we report standard deviations with time measurements.

Participation From the 13 systems participating to OAEI this year, 10 systems participated in this track. A few of these systems encountered problems:

- RSDLWB on cose
- OMReasoner on dog

We did not investigate these problems. We tried another test with many more ontologies an all matchers worked but AML.

Compliance Table 4 presents the harmonic means of precision, F-measure and recall for the test suites for all the participants, along with their confidence-weighted values. It also shows measures provided by edna, a simple edit distance algorithm on labels which is used as a baseline.

Some systems have had constant problems with the most strongly altered tests to the point of not outputing results: LogMap-C, LogMap, MaasMatch. Problems were also encountered to a smaller extent by XMap2. OMReasoner failed to return any answer on dog, and RSDLWB on cose.

Concerning F-measure results, the AOTL system seems to achieve the best results before RSDLWB. AOTL is also well balanced: it always achieve more than 50% recall with still a quite high precision. RSDLWD is slightly better than AOTL on two tests but

Matcher	biblio			cose			dog		
	Prec.	F-m.	Rec.	Prec.	F-m.	Rec.	Prec.	F-m.	Rec.
edna	.35(.58)	.41(.54)	.50	.44(.72)	.47(.59)	.50	.50(.74)	.50(.60)	.50
AML	.92(.94)	.55(.56)	.39	.46(.59)	.46(.51)	.46(.45)	.98(.96)	.73(.71)	.58(.57)
AOT	.80(.90)	.64(.67)	.53	.69(.84)	.58(.63)	.50	.62(.77)	.62(.68)	.61
AOTL	.85(.89)	.65(.66)	.53	.94(.95)	.65(.65)	.50	.97	.74(.75)	.60
LogMap	.40(.40)	.40(.39)	.40(.37)	.38(.45)	.41(.40)	.45(.37)	.96(.91)	.15(.14)	.08(.07)
LogMap-C	.42(.41)	.41(.39)	.40(.37)	.39(.45)	.41(.40)	.43(.35)	.98(.92)	.15(.13)	.08(.07)
LogMapLite	.43	.46	.50	.37	.43	.50	.86	.71	.61
MaasMatch	.97	.56	.39	.98	.48	.31	.92	.55	.39
OMReasoner	.73	.59	.50	.08	.14	.50	*	*	*
RSDLWB	.99	.66	.50	*	*	*	.99	.75	.60
XMap2	1.0	.57	.40	1.0	.28	.17	1.0	.32	.20

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

did not provide results on the third one. AOT is a close follower of AOTL. AML had very good results on dog and OMReasoner on biblio. The three systems showing the best performances at benchmarks (AOT, AOTL and RSDLWD) also preformed systematicly worse than other systems (AML, LogMap, XMap) at other tasks. This may reveal some degree of overfitting... either of the former to benchmarks, or of the latter to the other tests.

In general, results of the best matchers are largely lower than those of the best matchers in the previous year.

We can consider that we have high-precision matchers (XMap2: 1.0, RSDLWB: .99, MaasMatch: .92-.98; AML: (.46)-.98). LogMap-C, LogMap achieve also very high precision in dog (their other bad precision are certainly due to LogMap returning matched instances which are not in reference alignments). Of these high-precision matchers, RSDLWB is remarkable since it achieves a 50% recall (when it works).

The recall of systems is generally high with figures around 50% but this may be due to the structure of benchmarks.

Confidence-weighted measures reward systems able to provide accurate confidence values. Using confidence-weighted F-measures usually increase F-measure of systems showing that they are able to provide a meaningful assessment of their correspondences. The exception to this rule is LogMap whose weighted values are lower. Again, this may be due to the output of correspondences out of the ontology namespace or instance correspondences.

speed Table 5 provides the average time and standard deviation and F-measure point provided per second by matchers. The F-measure point provided per second shows that efficient matchers are XMap2 and LogMapLite followed by AML (these results are consistent on cose and dog, biblio is a bit different but certainly due to errors reported above). The time taken by systems on the two first test sets is very stable (and short); it is longer and less stable on the larger dog test set.

Matcher	biblio time	biblio stdev	biblio F-m./s.	cose time	cose stdev	cose F-m./s.	dog time	dog stdev	dog F-m./s.
AML	48.96	±1.21%	1.12	140.29	±0.98%	0.33	1506.16	±5.42%	0.05
AOT	166.91	±1.11%	0.38	194.02	±0.68%	0.30	10638.27	±0.77%	0.01
AOTL	741.98	±1.13%	0.09	386.18	±1.94%	0.17	18618.60	±1.44%	0.00
LogMap	106.68	±0.84%	0.37	123.44	±1.45%	0.33	472.31	±15.67%	0.03
LogMap-C	158.36	±0.53%	0.26	188.30	±1.22%	0.22	953.56	±18.94%	0.02
LogMapLite	61.43	±1.06%	0.75	62.67	±1.48%	0.69	370.32	±24.51%	0.19
MaasMatch	122.50	±2.24%	0.46	392.43	±1.78%	0.12	7338.92	±1.85%	0.01
OMReasoner	60.01	±0.43%	0.98	98.17	±0.91%	0.14	331.65	±59.35%	*
RSDLWB	86.22	±2.03%	0.77	*	*	*	14417.32	±1.98%	0.01
XMap2	68.67	±0.95%	0.83	31.39	±38.99%	0.89	221.83	±55.44%	0.14

Table 5. Aggregated benchmark results: Time (in second), standard deviation on time and points of F-measure per second spent on the three data sets (*: uncompleted results).

Comparison Figure 1 shows the triangle graphs for the three tests. It confirms the impressions above: systems are very precision-oriented but AOT which stands in the middle of the graph. AOTL has, in general, good results.

3.3 Conclusions

This year, matcher performance has been lower than in previous years, even on the genuine biblio dataset. The systems are able to process the test set without problem, even if some of them return many empty alignments. They are, as usual, very oriented towards precision at the expense of recall.

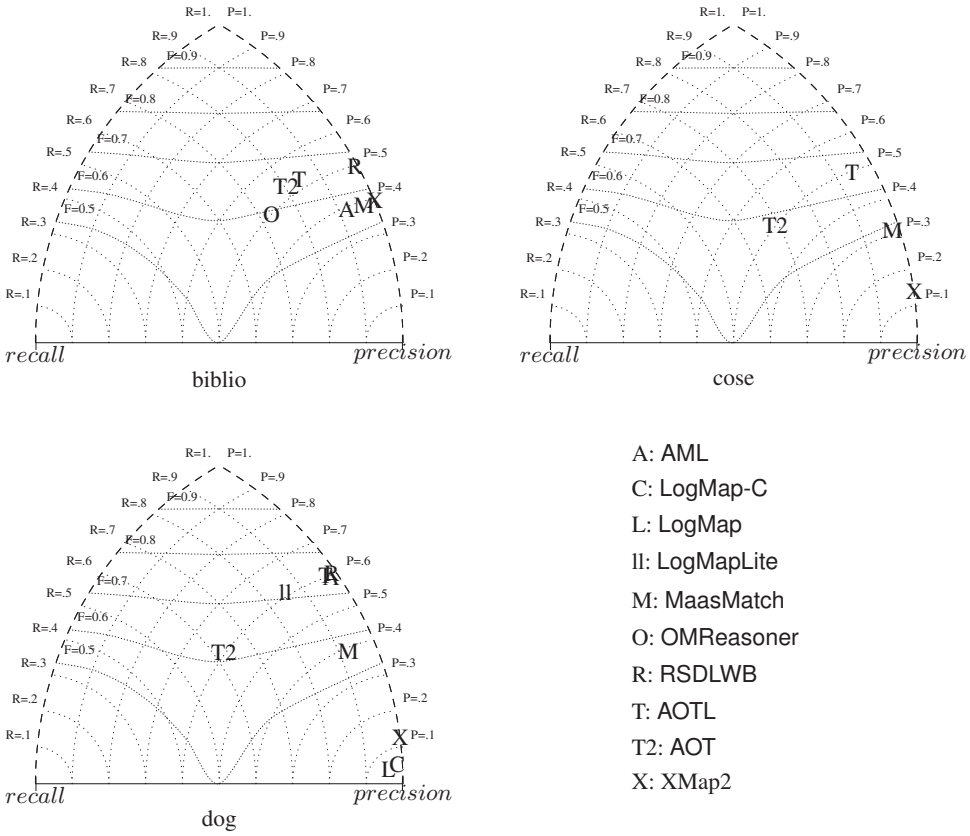


Fig. 1. Triangle view on the three benchmark data sets (non present systems have too low F-measure).

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[7] and the anatomy of the mouse[8]. This data set has been used since 2007 with some improvements over the years.

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 recall+ measure 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.

As last year, we run the systems on a server with 3.46 GHz (6 cores) and 8GB RAM allocated to the matching systems. Further, we used the SEALS client to execute our evaluation. However, we slightly changed the way precision and recall are computed, i.e., the results generated by the SEALS client vary in some cases by 0.5% compared to the results presented below. In particular, we removed trivial correspondences in the `oboInOwl` namespace such as

```
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 6, we analyze all participating systems that could generate an alignment in less than ten hours. The listing comprises 10 entries. There were 2 systems which participated with different versions. These are AOT with versions AOT and AOTL, LogMap with four different versions LogMap, LogMap-Bio, LogMap-C and a lightweight version, LogMapLite, that uses only some core components. In addition to LogMap and LogMapLite, 3 more systems which participated in 2013 and now participated with new versions (AML, MaasMatch, XMap). For more details, we refer the reader to the papers presenting the systems. Thus, 10 different systems generated an alignment within the given time frame. There were four participants (InsMT, InsMTL, OMReasoner and RiMOM-IM) that threw an exception or produced an empty alignment and are not considered in the evaluation.

We have 6 systems which finished in less than 100 seconds, compared to 10 systems in OAEI 2013 and 8 systems in OAEI 2012. This year we have 10 out of 13 systems which generated results which is comparable to last year when 20 out of 24 systems

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

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

Matcher	Runtime	Size	Precision	F-measure	Recall	Recall+	Coherent
AML	28	1478	0.956	0.944	0.932	0.822	√
LogMap-Bio	535	1547	0.888	0.897	0.906	0.752	√
XMap	22	1370	0.940	0.893	0.850	0.606	√
LogMap	12	1398	0.918	0.881	0.846	0.595	√
LogMapLite	5	1148	0.962	0.829	0.728	0.290	-
MaasMatch	49	1187	0.914	0.803	0.716	0.248	-
LogMap-C	22	1061	0.975	0.802	0.682	0.433	√
StringEquiv	-	946	*1.000*	0.770	0.620	0.000	-
RSDLWB	1337	941	0.978	0.749	0.607	0.01	-
AOT	896	2698	0.436	0.558	0.775	0.405	-
AOTL	2524	167	0.707	0.140	0.078	0.010	-

Table 6. 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.

generated results within the given time frame. The top systems in terms of runtimes are LogMap, XMap and AML. Depending on the specific version of the systems, they require between 5 and 30 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 6 also shows the results for precision, recall and F-measure. In terms of F-measure, the top ranked systems are AML, LogMap-Bio, LogMap and XMap. The latter two generate similar alignments. The results of these four systems are at least as good as the results of the best systems in OAEI 2007-2010. AML has the highest F-measure up to now. Other systems in earlier years that obtained an F-measure that is at least as good as the fourth system this year are AgreementMaker (predecessor of AML) (2011, F-measure: 0.917), GOMMA-bk (2012/2013, F-measure: 0.923/0.923), YAM++ (2012/2013, F-measure 0.898/0.905), and CODI (2012, F-measure: 0.891).

This year we have 7 out of 10 systems which achieved an F-measure that is higher than the baseline which is based on (normalized) string equivalence (StringEquiv in the table). This is a better result (percentage-wise) than the last year but still lower than in OAEI 2012 when 13 out of 17 systems produced alignments with F-measure higher than the baseline. Both systems, XMap and MaasMatch, which participated in the last year and had results below the baseline, achieved better results than the baseline this year.

Moreover, nearly all systems find many non-trivial correspondences. Exceptions are RSDLWB and AOTL that generate an alignment that is quite similar to the alignment generated by the baseline approach.

There are 5 systems which participated in the last year, AML, LogMap, LogMapLite, MaasMatch and XMap. From these systems LogMap and LogMapLite achieved identical results as last year, while AML, MaasMatch and XMap improved their results. MaasMatch and XMap showed a considerable improvement. In the case of MaasMatch, its precision was improved from 0.359 to 0.914 (and the F-measure from 0.409 to 0.803) while XMap which participated with two versions in the last year increased its precision

from 0.856 to 0.94 (and F-measure from 0.753 to 0.893) compared to the XMapSig version which achieved a better F-measure last year.

A positive trend can be seen when it comes to coherence of alignments. Last year only 3 systems out of 20 produced a coherent alignment while this year half of the systems produced coherent alignment.

4.3 Conclusions

This year 14 systems participated in the anatomy track out of which 10 produced results. This is a significant decrease in the number of participating systems. However, the majority of the systems which participated in the last year significantly improved their results.

As last year, we have witnessed a positive trend in runtimes as all the systems which produced an alignment finished execution in less than an hour. Same as the last year, the AML system set the top result for the anatomy track by improving the result from the last year. The AML system improved in terms of all measured metrics.

5 Conference

The conference test case introduces matching several moderately expressive ontologies. Within this test case, participant alignments were evaluated against reference alignments (containing merely equivalence correspondences) and by using logical reasoning. The evaluation has been performed with the SEALS infrastructure.

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[9].

The main features of this test case are:

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

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

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

5.2 Results

We provide results in terms of $F_{0.5}$-measure, F_1-measure and F_2-measure, comparison with baseline matchers and results from previous OAEI editions, precision/recall triangular graph and coherency evaluation.

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

These reference alignments have been generated as a transitive closure computed on the original reference alignments. In order to obtain a coherent result, conflicting correspondences, i.e., those causing unsatisfiability, have been manually inspected and removed by evaluators. 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.	Size	Inc. Al.	Inc-dg
AML	0.8	0.74	0.67	0.61	0.58	10.952	0	0.0%
LogMap	0.76	0.7	0.63	0.57	0.54	10.714	0	0.0%
LogMap-C	0.78	0.71	0.62	0.56	0.52	10.238	0	0.0%
XMap	0.82	0.7	0.57	0.48	0.44	8.143	0	0.0%
edna	*0.73*	*0.64*	*0.55*	*0.48*	*0.44*			
AOT*	0.75	0.65	0.55	0.47	0.43	59.167	18	40.4%
RSDLWB	0.76	0.65	0.54	0.46	0.42	8.333	4	2.5%
LogMapLite	0.68	0.62	0.54	0.48	0.45	9.905	7	5.4%
OMReasoner	0.77	0.66	0.54	0.46	0.42	8.095	4	2.5%
StringEquiv	*0.76*	*0.64*	*0.52*	*0.43*	*0.39*			
AOTL	0.73	0.62	0.51	0.43	0.39	14.667	17	15.1%
MaasMatch*	0.52	0.51	0.5	0.5	0.49	33	19	21.0%

Table 7. 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). Average size of alignments, number of incoherent alignments and average degree of incoherence. The mark * is added when we only provide lower bound of the degree of incoherence due to the combinatorial complexity of the problem.

Table 7 shows the results of all participants with regard to the reference alignment. $F_{0.5}$-measure, F_1-measure and F_2-measure are computed for the threshold that provides the highest average F_1-measure. F_1 is the harmonic mean of precision and recall where both are equally weighted; F_2 weights recall higher than precision and $F_{0.5}$ weights precision higher than recall. The matchers shown in the table are ordered according to their highest average F_1-measure. 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 previously used *baseline2* in the conference track; *StringEquiv* is used within the anatomy test case. These baselines divide matchers into three groups. Group

1 consists of matchers (AML, LogMap, LogMap-C, XMap and AOT) having better (or the same) results than both baselines in terms of highest average F_1-measure. Group 2 consists of matchers (RSDLWB, LogMapLite and OMReasoner) performing better than baseline *StringEquiv*. Other matchers (AOTL and MaasMatch) performed slightly worse than both baselines.

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.

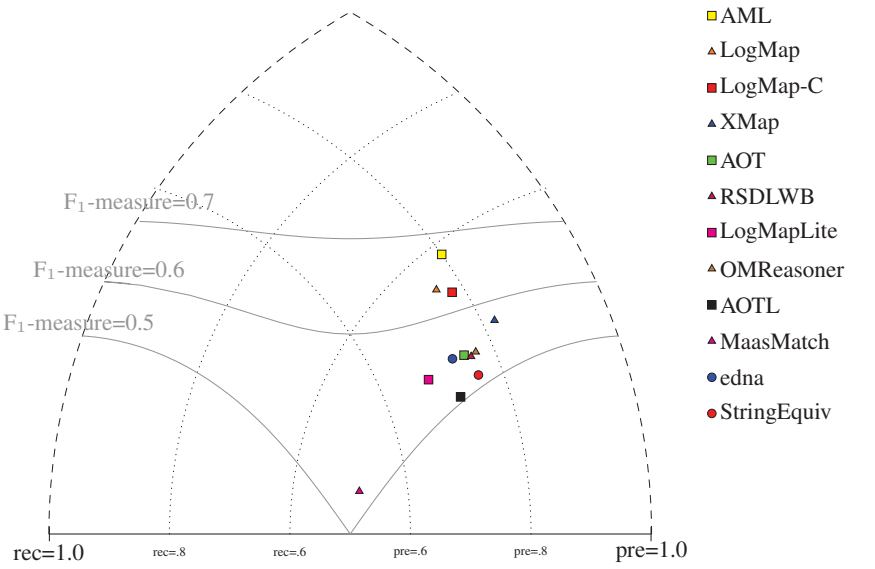


Fig. 2. Precision/recall triangular graph for the conference test case. Dotted lines depict level of precision/recall while values of F_1-measure are depicted by areas bordered by corresponding lines F_1-measure=0.[5|6|7].

Comparison with previous years Five matchers also participated in this test case in OAEI 2013. The largest improvement was achieved by MaasMatch (precision from .27 to .52, while recall decreased from .53 to .49), AML (precision decreased from .82 to .80, but recall increased from .51 to .58) and XMap (precision from .68 to .82, whereas recall remains the same, .44).

Runtimes We measured the total time of generating 21 alignments. It was executed on a laptop under Ubuntu running on Intel Core i5, 2.67GHz and 8GB RAM except MaasMatch run which was run on Intel Core i7, 2.10GHz x 4 and 16GB RAM. This year all matchers finished all 21 testcases within 70 seconds. Four matchers finished all 21 test cases within 16 seconds (OMReasoner: 10s, LogMapLite: 11s, AML: 14s and AOT: 16s). Next, five matchers needed less than 1 minute (LogMap: 26s, XMap: 26s, RSDLWB: 36s, LogMap-C: 44s, AOTL: 45s). Finally, one matcher (MaasMatch) needed 69 seconds to finish all 21 test cases.

In conclusion, regarding performance we can see (clearly from Figure 2) that almost all participants managed to achieve a higher performance than baseline matcher. Three matchers (AML, LogMap and LogMap-C) exceeded a 0.6 F1-measure and all other matchers are above 0.5. On the other side no matcher achieved a 0.7 F1-measure. Regarding runtime, the four fastest matchers this year managed to be faster than the fastest matcher last year (measured on the same machine) and no matcher needed more than 70 seconds which is much faster than last year (40 minutes).

Evaluation based on alignment coherence As in the previous years, we apply the Maximum Cardinality measure to evaluate the degree of alignment incoherence. Details on this measure and its implementation can be found in [23].

We computed the average for all 21 test cases of the conference track for which there exists a reference alignment. In two cases (marked with an asterisk) we could not compute the exact degree of incoherence due to the combinatorial complexity of the problem, however we were still able to compute a lower bound for which we know that the actual degree is not lower.

The systems AML, LogMap (excluding LogMapLite, where reasoning option is disabled), and XMap generate coherent alignments. However, these systems generated coherent alignments already in 2013. The other systems generate results with highly varying degree of incoherence. The degree of incoherence is correlated with the size of the generated alignments. This can be expected because smaller alignments are usually more precise and logical conflicts will occur only rarely. However, there are systems with relatively small alignments that cannot ensure coherence (e.g., OMReasoner and RSDLWB). Overall, the field has not improved compared to last year with respect to generating coherent alignments respecting the logical constraints implied by the axioms of the matched ontologies.

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 [15]) between the integrated ontologies, it is relatively straightforward to extract them from the information provided in the distribution files (see [18] 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 [18].

Since alignment coherence is an aspect of ontology matching that we aim to promote in the Large BioMed track, in previous editions we provided coherent reference alignments by refining the UMLS mappings using Alcomo (alignment) debugging system [23], LogMap's (alignment) repair facility [17], or both [19].

However, concerns were raised about the validity and fairness of applying automated alignment repair techniques to make reference alignments coherent [27]. 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 [27].

Thus, for this year's edition of the largebio track 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 [29], 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 2014 campaign is summarised as follows:

- FMA-NCI reference alignment: 2,686 "=" mappings, 338 "?" mappings
- FMA-SNOMED reference alignment: 6,026 "=" mappings, 2,982 "?" mappings
- SNOMED-NCI reference alignment: 17,210 "=" mappings, 1,634 "?" mappings

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 largebio test case, 11 out of 14 participating systems have been able to cope with at least one of the tasks of the largebio test case. It is surprising, but for the first year the largebio track had the largest participation with respect to the other tracks.

System	FMA-NCI		FMA-SNOMED		SNOMED-NCI		Average	#
	Task 1	Task 2	Task 3	Task 4	Task 5	Task 6		
LogMapLite	5	44	13	90	76	89	53	6
XMap	17	144	35	390	182	490	210	6
LogMap	14	106	63	388	263	917	292	6
AML	27	112	126	251	831	497	307	6
LogMap-C	81	289	119	571	2,723	2,548	1,055	6
LogMap-Bio	975	1,226	1,060	1,449	1,379	2,545	1,439	6
OMReasoner	82	36,369	691	-	5,206	-	10,587	4
MaasMatch	1,460	-	4,605	-	-	-	3,033	2
RSDLWB	2,216	-	-	-	-	-	2,216	1
AOT	9,341	-	-	-	-	-	9,341	1
AOTL	20,908	-	-	-	-	-	20,908	1
# Systems	**11**	**7**	**8**	**6**	**7**	**6**	**4,495**	**45**

Table 8. System runtimes (s) and task completion.

RiMOM-IM, InsMT and InsMTL are systems focusing in the instance matching track and they did not produce any alignment for the largebio track.

Regarding the use of background knowledge, LogMap-Bio uses BioPortal as mediating ontology provider, that is, it retrieves from BioPortal the most suitable top-5 ontologies for the matching task.

6.3 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 [25] 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 [20].

In this OAEI edition, only two systems have shown alignment repair facilities, namely: AML and LogMap (including LogMap-Bio and LogMap-C variants). Tables 9-12 (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.

6.4 Runtimes and task completion

Table 8 shows which systems were able to complete each of the matching tasks in less than 10 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
AML	27	2,690	0.96	0.93	0.90	2	0.02%
LogMap	14	2,738	0.95	0.92	0.90	2	0.02%
LogMap-Bio	975	2,892	0.91	0.92	0.92	467	4.5%
XMap	17	2,657	0.93	0.89	0.85	3,905	38.0%
LogMapLite	5	2,479	0.97	0.89	0.82	2,103	20.5%
LogMap-C	81	2,153	0.96	0.83	0.72	2	0.02%
MaasMatch	1,460	2,981	0.81	0.82	0.84	8,767	85.3%
Average	3,193	2,287	0.91	0.76	0.70	2,277	22.2%
AOT	9,341	3,696	0.66	0.75	0.85	8,373	81.4%
OMReasoner	82	1,362	0.99	0.63	0.47	56	0.5%
RSDLWB	2,216	728	0.96	0.38	0.24	22	0.2%
AOTL	20,908	790	0.90	0.38	0.24	1,356	13.2%

Task 2: whole FMA and NCI ontologies							
System	Time (s)	# Corresp.	Scores			Incoherence	
			Prec.	F-m.	Rec.	Unsat.	Degree
AML	112	2,931	0.83	0.84	0.86	10	0.007%
LogMap	106	2,678	0.86	0.83	0.81	13	0.009%
LogMap-Bio	1,226	3,412	0.72	0.79	0.87	40	0.027%
XMap	144	2,571	0.83	0.79	0.75	9,218	6.3%
Average	5,470	2,655	0.82	0.77	0.75	5,122	3.5%
LogMap-C	289	2,124	0.88	0.75	0.65	9	0.006%
LogMapLite	44	3,467	0.67	0.74	0.82	26,441	18.1%
OMReasoner	36,369	1,403	0.96	0.63	0.47	123	0.084%

Table 9. Results for the FMA-NCI matching problem.

The last column reports the number of tasks that a system could complete. For example, 6 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.

6.5 Results for the FMA-NCI matching problem

Table 9 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.

LogMap-Bio and AML provided the best results in terms of both Recall and F-measure in Task 1 and Task 2, respectively. OMReasoner provided the best results in terms of precision, although its recall was below average. From the last year participants, XMap and MaasMatch improved considerably their performance with respect to both runtime and F-measure. AML and LogMap obtained again very good results. LogMap-Bio improves LogMap's recall in both tasks, however precision is damaged specially in Task 2.

Task 3: small FMA and SNOMED fragments							
System	Time (s)	# Corresp.	Scores			Incoherence	
			Prec.	F-m.	Rec.	Unsat.	Degree
AML	126	6,791	0.93	0.82	0.74	0	0.0%
LogMap-Bio	1,060	6,444	0.93	0.81	0.71	0	0.0%
LogMap	63	6,242	0.95	0.80	0.70	0	0.0%
XMap	35	7,443	0.86	0.79	0.74	13,429	56.9%
LogMap-C	119	4,536	0.96	0.66	0.51	0	0.0%
MaasMatch	4,605	8,117	0.65	0.66	0.67	21,946	92.9%
Average	839	5,342	0.87	0.64	0.55	4,578	19.4%
LogMapLite	13	1,645	0.97	0.34	0.21	773	3.3%
OMReasoner	691	1,520	0.71	0.26	0.16	478	2.0%

Task 4: whole FMA ontology with SNOMED large fragment							
System	Time (s)	# Corresp.	Scores			Incoherence	
			Prec.	F-m.	Rec.	Unsat.	Degree
AML	251	6,192	0.89	0.75	0.65	0	0.0%
LogMap	388	6,141	0.83	0.71	0.62	0	0.0%
LogMap-Bio	1,449	6,853	0.76	0.70	0.65	0	0.0%
Average	523	5,760	0.79	0.62	0.54	11,823	5.9%
LogMap-C	571	4,630	0.85	0.61	0.48	98	0.049%
XMap	390	8,926	0.56	0.59	0.63	66,448	33.0%
LogMapLite	90	1,823	0.85	0.33	0.21	4,393	2.2%

Table 10. Results for the FMA-SNOMED matching problem.

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, AOT, AOTL, RSDLWB and MaasMatch could not complete Task 2. The first three did not finish in less than 10 hours while MaasMatch rose an "out of memory" exception.

6.6 Results for the FMA-SNOMED matching problem

Table 10 summarizes the results for the tasks in the FMA-SNOMED matching problem. AML provided the best results in terms of F-measure on both Task 3 and Task 4. AML also provided the best Recall and Precision in Task 3 and Task 4, respectively; while LogMapLite provided the best Precision in Task 3 and LogMap-Bio the best Recall in Task 4.

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 LogMapLite and XMap in terms of precision. Furthermore, AOT, AOTL, RSDLWB could not complete neither Task 3 nor Task 4 in less than 10 hours. MaasMatch rose an "out of memory" exception in Task 4, while OMReasoner could not complete Task 4 within the allowed time.

Task 5: small SNOMED and NCI fragments							
System	Time (s)	# Corresp.	Scores			Incoherence	
			Prec.	F-m.	Rec.	Unsat.	Degree
AML	831	14,131	0.92	0.81	0.72	≥0	≥0.0%
LogMap-Bio	1,379	14,360	0.88	0.79	0.71	≥23	≥0.031%
LogMap	263	14,011	0.89	0.78	0.70	≥23	≥0.031%
XMap	182	14,223	0.85	0.75	0.66	≥65,512	≥87.1%
Average	1,522	12,177	0.91	0.72	0.61	≥23,078	≥30.7%
LogMapLite	76	10,962	0.95	0.71	0.57	≥60,426	≥80.3%
LogMap-C	2,723	10,432	0.91	0.67	0.53	≥0	≥0.0%
OMReasoner	5,206	7,120	0.98	0.55	0.38	≥35,568	≥47.3%

Task 6: whole NCI ontology with SNOMED large fragment							
System	Time (s)	# Corresp.	Scores			Incoherence	
			Prec.	F-m.	Rec.	Unsat.	Degree
AML	497	12,626	0.91	0.76	0.65	≥0	≥0.0%
LogMap-Bio	2,545	12,507	0.85	0.70	0.60	≥37	≥0.020%
LogMap	917	12,167	0.86	0.70	0.59	≥36	≥0.019%
XMap	490	12,525	0.84	0.69	0.58	≥134,622	≥71.1%
Average	1,181	12,024	0.86	0.69	0.57	≥47,578	≥25.1%
LogMapLite	89	12,907	0.80	0.66	0.57	≥150,776	≥79.6%
LogMap-C	2,548	9,414	0.88	0.61	0.46	≥1	≥0.001%

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

6.7 Results for the SNOMED-NCI matching problem

Table 11 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, while OMReasoner provided the best results in terms of precision. Task 6 was completely dominated by AML.

As in the previous matching problems, efficiency decreases as the ontology size increases. Furthermore, AOT, AOTL, RSDLWB could not complete Task 5 nor Task 6 in less than 10 hours. MaasMatch rose a "stack overflow" exception in Task 5 and an "out of memory" exception in Task 6, while OMReasoner could not complete Task 6 within the allocated time.

6.8 Summary results for the top systems

Table 12 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 was a step ahead and obtained the best average Recall and F-measure, and the second best average Precision. LogMap-C obtained the best average Precision while LogMap-Bio obtained the second best average Recall.

System	Total Time (s)	Average			
		Prec.	F-m.	Rec.	Inc. Degree
AML	1,844	0.91	0.82	0.75	0.004%
LogMap	1,751	0.89	0.79	0.72	0.013%
LogMap-Bio	8,634	0.84	0.78	0.74	0.8%
XMap	1,258	0.81	0.75	0.70	48.7%
LogMap-C	6,331	0.91	0.69	0.56	0.013%
LogMapLite	317	0.87	0.61	0.53	34.0%

Table 12. Summary results for the top systems.

Regarding alignment incoherence, AML also computed, on average, the correspondence sets leading to the smallest number of unsatisfiable classes. LogMap variants also obtained very good results in terms of alignment coherence.

Finally, LogMapLite was the fastest system. The rest of the tools were also very fast and only needed between 21 and 144 minutes to complete all 6 tasks.

6.9 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 6 systems completed all matching tasks with very competitive results. Furthermore, 11 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) the size of the input ontologies should not significantly affect efficiency, 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 9-12, even highly precise alignment sets may lead to a huge number of unsatisfiable classes (e.g. LogMapLite and OMReasoner 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 2014 have shown to successfully use 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 [23], the repair module of LogMap (LogMap-Repair) [17] or the repair module of AML [29], which have shown to work well in practice [19, 16].

7 MultiFarm

The MultiFarm data set [24] 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 Conference track ontologies (cmt, conference, confOf, iasted, sigkdd, ekaw and edas), into 8 languages: Chinese, Czech, Dutch, French, German, Portuguese, Russian, and Spanish (+ English). These translations result in 36 pairs of languages. 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. Hence, MultiFarm is composed of 36×49 matching tasks.

7.1 Experimental setting

For the 2014 campaign, part of the data set has been used for a kind of blind evaluation. This subset include all the pairs of matching tasks involving the edas and ekaw ontologies (resulting in 36×24 matching tasks), which were not used in previous campaigns[10]. We refer to evaluation as *edas and ekaw based evaluation* in the following. Participants were able to test their systems on the freely available sub-set of matching tasks (*open evaluation*) (including reference alignments), available via the SEALS repository, which is composed of 36×25 tasks.

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

This year, only 3 systems (out of 14 participants, see Table 2) use specific cross-lingual[11] methods: AML, LogMap and XMap. This number drastically decreased with respect to the last two campaigns: 7 systems in 2013 and 7 in 2012. All of them integrate a translation module in their implementations. LogMap uses Google Translator API and pre-compiles a local dictionary in order to avoid multiple accesses to the Google server within the matching process. AML and XMap use Microsoft Translator, and AML adopts the same strategy of LogMap computing a local dictionary. The translation step is performed before the matching step itself.

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, under Linux ProxMox 2 (Debian). With respect to runtime, we compare all systems on the basis of the open data set and their runtimes

[10] In fact, this subset was, two years ago, by error, available on the MultiFarm web page. Since that, we have removed it from there and it is not available as well for the participants via the SEALS repositories. However, we cannot guarantee that the participants have not used this data set for their tests.

[11] As already reported in the last campaign, we have revised the definitions of multilingual and cross-lingual matching. Initially, as reported in [24], MultiFarm was announced as a benchmark for multilingual ontology matching, i.e., *multilingual* in the sense that we have a set of ontologies in 8 languages. However, it is more appropriate to use the term *cross-lingual* ontology matching. Cross-lingual ontology matching refers to the matching cases where each ontology uses a different natural language (or a different set of natural languages) for entity naming, i.e., the intersection of sets is empty. It is the case of matching tasks in MultiFarm.

can be found in Table 13. All measurements are based on a single run. Systems not listed in Table 13 have not been executed in this track – InsMT, InsMTL, RiMOM-IM (dedicated to the IM track) and LogMapBio (dedicated to LargeBio track) – or have encountered problems to parse the ontologies (OMReasoner). Some exceptions were observed for MaasMatch, which was not able to be executed under the same setting than the other systems. Thus, we do not report on execution time for this system.

We can observe large differences between the time required for a system to complete the 36×25 matching tasks. While AML takes around 8 minutes, XMap requires around 24 hours. Under a same setting LogMap took around 18 minutes in 2013 and around 2 hours this year. This is due to the fact that the local dictionaries are incomplete and accesses to Google Translator server have to be performed for some pairs, what may explain the increase in the execution time.

7.3 Evaluation results

Open evaluation results Before discussing the results for the *edas and ekaw based evaluation*, we present the aggregated results for the open subset of MultiFarm, for the test cases of type (i) and (ii) (Table 13). The results have been computed using the Alignment API 4.6. We did not distinguish empty and erroneous alignments. We observe significant differences between the results obtained for each type of matching task, specially in terms of precision, for all systems, with lower differences in terms of recall. As expected, all systems implementing specific cross-lingual techniques generate the best results for test cases of type (i). A similar behavior has also been observed for the tests cases of type (ii), even if the specific strategies could have less impact due to the fact that the identical structure of the ontologies could also be exploited instead by the other systems. For cases of type (i), 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 (what is the case for both types of tasks).

		Type (i)				Type (ii)			
System	Time	Size	Prec.	F-m.	Rec.	Size	Prec.	F-m.	Rec.
AML	8	11.40	.57	.54	.53	54.89	.95	.62	.48
LogMap	128	5.04	.80	.40	.28	36.07	.94	.41	.27
XMap	1455	110.79	.31	.35	.43	67.75	.76	.50	.40
AOT	21	106.29	.02	.04	.17	109.79	.11	.12	.12
AOTL	48	1.86	.10	.03	.02	2.65	.27	.02	.01
LogMap-C	25	1.30	.15	.04	.02	3.52	.31	.02	.01
LogMapLite	6	1.73	.13	.04	.02	3.65	.25	.02	.01
MaasMatch	-	3.16	.27	.15	.10	7.71	.52	.10	.06
RSDLWB	18	1.31	.16	.04	.02	2.41	.34	.02	.01

Table 13. MultiFarm aggregated results per matcher (average), for each type of matching task – different ontologies (i) and same ontologies (ii). Time is measured in minutes (time for completing the 36×25 matching tasks). Size indicates the average of the number of generated correspondences for each test type.

With respect to the specific pairs of languages for test cases of type (i), for the sake of brevity, we do not detail them here. The reader can refer to the OAEI results web page for detailed results for each of the 36 pairs of languages. As expected and already reported above, systems that apply specific strategies to match ontology entities described in different natural languages outperform all other systems. As already observed for the best system last year (YAM++), the best results in terms of F-measure for AML has been observed for the pairs involving Czech – cz-en (.63), cz-ru (.63), cz-es (.61), cz-nl (.60) – followed of pairs involving English and Russian – en-ru (.60). In the case of LogMap, for pairs involving English, Spanish – en-es (.61) – and Czech – cz-en (.60) – it generates its best scores, followed by en-pt (.56) and de-en (.56). As AML, top F-measure results for XMap are observed for the pair involving Czech – cz-es (.50), cz-fr (.47), cz-pt (.46). However, when dealing with cases of type (ii), these systems generate best results for the pairs involving English, French, Portuguese and Spanish (including Dutch for LogMap).

For non-specific systems, most of them cannot deal with Chinese and Russian languages. All of them generate their best results for the pairs es-pt and de-en: AOT (es-pt .10), AOTL (de-en .19), LogMap-C (de-en .20), LogMapLite (es-pt .23) MaasMatch (de-en .37) and RSDLWB (es-pt .23), followed by es-fr, en-es and fr-nl. These systems take advantage of similarities in the vocabulary for these languages in the matching task, in the absence of specific strategies. A similar result has been observed last year for non-specific systems, where 7 out of 10 cross-lingual systems generated their best results for the pair es-pt, followed by the pair de-en. On the other hand, although it is likely harder to find correspondences between cz-pt than es-pt, for some systems Czech is present in the top-5 F-measure (cz-pt, for LogMap-C, LogMapLite and RSDLWB or cz-es for AOTL, LogMapLite and RSDLWB). It can be explained by the specific way systems combine their internal matching techniques (ontology structure, reasoning, coherence, linguistic similarities, etc).

Edas and Ekaw based evaluation In the first year of MultiFarm evaluation, we have used a subset of the whole data set, where we omitted the ontologies edas and ekaw, and suppressed the test cases where Russian and Chinese were involved. Since 2012, we have included Russian and Chinese translations, and this year we have included edas and ekaw in a (pseudo) blind setting, as explained above. We evaluate this subset on the systems implementing specific cross-lingual strategies. The tools run in the SEALS platform using locally stored ontologies. Table 14 presents the results for AML and LogMap. Using this setting, XMap has launched exceptions for most pairs and its results are not reported for this subset. These internal exceptions were due to the fact that the system exceeded the limit of accesses to the translator and could not generate any translation for most pairs. While AML includes in its local dictionaries the automatic translations for the two ontologies, it is not the case for LogMap (real blind case). This can explain the similar results obtained by AML in both settings. However, LogMap has encountered many problems for accessing Google translation server from our server, what explain the decrease in its results and the increase in runtime (besides the fact that this data set is slightly bigger than the open data set in terms of ontology elements).

Overall, for cases of type (i) – remarking the particular case of AML – the systems maintained their performance with respect to the open setting.

		Type (i)				Type (ii)			
System	Time	Size	Prec.	F-m.	Rec.	Size	Prec.	F-m.	Rec.
AML	14	12.82	.55	.47	.42	64.59	.94	.62	.46
LogMap	219	5.21	.77	.33	.22	71.13	.19	.14	.11

Table 14. 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, in minutes, for completing the 36×24 matching task.

Comparison with previous campaigns In the first year of evaluation of MultiFarm (2011.5 campaign), 3 participants (out of 19) used specific techniques. In 2012, 7 systems (out of 24) implemented specific techniques for dealing with ontologies in different natural languages. We had the same number of participants in 2013. This year, none of these systems has participated. However, we count with 3 systems implementing cross-lingual strategies (AML, LogMap and XMap), as extensions of versions participating in previous campaigns. Comparing 2013 and 2012 F-measure results (on the same basis - type (ii)), this year AML (.54) outperformed the best system in 2013 and 2012 – YAM++ (.40) – while LogMap (.40) had similar results. In overall, we observe a global improvement in performance this year for systems implementing specific matching strategies. With respect to non-specific systems, MaasMatch increased F-measure for tests of type (i) – from .01 up to .15 – and decreased that of cases (ii) – .29 to .10. Its good performance in (ii) may be explained by the implementation of new similarity aggregations reflecting similarity values even when few overlaps exist.

7.4 Conclusion

As we could expect, systems implementing specific methods for dealing with ontologies in different languages outperform non specific systems. However, since the first campaign MultiFarm is proposed, the absolute results are still not very good, if compared to the top results of the original Conference data set (approximatively 74% F-measure for the best matcher). Although only 3 systems have implemented specific strategies this year, in terms of overall results, one of them has outperformed the best systems in previous campaigns. However, the adopted strategies are rather limited to translations steps before the matching step itself. Again, all systems privilege precision rather than recall. Both in terms of matching strategies and results, there is still room for improvements. As future work, we plan to provide a new version of the data set, correcting as well some typos identified in the translations. We envisage as well to add the Italian translations as (real) blind evaluation.

8 Library

The library test case was established in 2012[12]. The test case consists of matching two real-world thesauri: The Thesaurus for the Social Sciences (TSS, maintained by GESIS) and the Standard Thesaurus for Economics (STW, maintained by ZBW). The reference alignment is based on a manually created alignment in 2006. As additional benefit from this test case, the reference alignment is constantly updated by the maintainers with the generated correspondences that are checked manually when they are not part of the reference alignment.[13]

8.1 Test data

Both thesauri used in this test case are comparable in many respects. They have roughly the same size (6,000 resp. 8,000 concepts), are both originally developed in German, are both translated into English, and, most important, despite being from two different domains, they have significant overlapping areas. Not least, both are freely available in RDF using SKOS.[14] To enable the participation of all OAEI matchers, an OWL version of both thesauri is provided, effectively by creating a class hierarchy from the concept hierarchy. Details are provided in the report of the 2012 campaign [1]. For the first time, we also created an OWL version containing SKOS annotations like preferred and alternative label as OWL annotations. As stated above, we updated the reference alignment with all correct correspondences found during the last campaigns. It now consists of 3161 correspondences.

8.2 Experimental setting

All matching processes have been performed on a Debian machine with one 2.4GHz core and 7GB RAM allocated to each system. The evaluation has been executed by using the SEALS infrastructure. Each participating system uses the OWL version, two systems make use of the additional SKOS annotations.

To compare the created alignments with the reference alignment, we use the Alignment API. For this evaluation, we only included equivalence relations (skos:exactMatch). We computed precision, recall and F_1-measure for each matcher. Moreover, we measured the runtime, the size of the created alignment, and checked whether a 1:1 alignment has been created. To assess the results of the matchers, we developed three straightforward matching strategies, using the original SKOS version of the thesauri:

- MatcherPrefDE: Compares the German lower-case preferred labels and generates a correspondence if these labels are completely equivalent.
- MatcherPrefEN: Compares the English lower-case preferred labels and generates a correspondence if these labels are completely equivalent.

[12] There has already been a library test case from 2007 to 2009 using different thesauri, as well as other thesaurus test cases like the food and the environment test cases.

[13] With the reasonable exception of XMapGen, which produces almost 40.000 correspondences.

[14] `http://www.w3.org/TR/skos-reference/`

– MatcherPref: Creates a correspondence, if either MatcherPrefDE or MatcherPrefEN or both create a correspondence.
– MatcherAllLabels: Creates a correspondence whenever at least one label (preferred or alternative, all languages) of an entity is equivalent to one label of another entity.

8.3 Results

Of all 12 participating matchers (or variants), 7 were able to generate an alignment within 8 hours. The results can be found in Table 15.

Matcher	Precision	F-Measure	Recall	Time (ms)	Size	1:1
AML*	0.82	0.80	0.78	68489	2983	-
MatcherPref	0.91	0.74	0.63	-	2190	-
AML	0.72	0.73	0.75	71070	3303	-
MatcherPrefDE	0.98	0.73	0.58	-	1885	-
MatcherAllLabels	0.61	0.72	0.89	-	4605	-
LogMap*	0.74	0.71	0.68	222668	2896	-
LogMap	0.78	0.71	0.65	73964	2642	-
LogMapLite	0.64	0.70	0.77	9329	3782	-
XMap2	0.51	0.65	0.89	12652823	5499	-
MatcherPrefEN	0.88	0.57	0.42	-	1518	-
MaasMatch	0.50	0.57	0.66	14641118	4117	x
LogMap-C	0.48	0.34	0.26	21859	1723	-
RSDLWB	0.78	0.07	0.04	32828314	155	x

Table 15. Results of the Library test case (ordered by F-measure).

The best systems in terms of F-measure are AML and LogMap. AML* and LogMap* are the matching systems performed on the OWL-dataset with SKOS annotations. For both systems, using this ontology version increases the F-measure up to 7% which shows that the additional information is useful. Except for AML, all systems are below the MatcherPrefDE and MatcherAllLabels strategies. A group of matchers including LogMap, LogMapLite, and XMap2 are above the MatcherPrefEN baseline. Compared to the evaluation conducted last year, the results are similar: The baselines with preferred labels are still very good and can only be beaten by one system. AML* has a better F-Measure than any other system before (4% increase compared to the best matcher of last year).

Like in previous years, an additional intellectual evaluation of the alignments established automatically was done by a domain expert to further improve the reference alignment. Since the competing ontology matching tools predominantly apply lexical approaches for matching the two vocabularies they foremost establish new correspondences on the character level. The main approaches that are applied here are Levenshtein distance or string recognition where character strings could consist of up to a whole part of a compound word, partly used as an adjective. Together with the three above described straightforward matching strategies, these character respectively string matching approaches lead to different types of systematic mismatches. Especially in the case of short terms, Levensthein distance could lead to wrong correspondences, e.g., “Ziege” (Eng. goat) and “Zeuge” (Eng. witness) or “Dumping”

(Eng. dumping) and “Doping” (Eng. doping). Mere string matching often leads to wrong correspondences. Typical cases include partial matchings at the beginning, in the middle, or at the end of a word, like “Monopson” (Eng. monopsony) and “Monotonie” (Eng. monotony), “Zession” (Eng. cession) and “Rezession” (Eng. recession), or “Rohrleitungsbau” (Eng. pipeline construction) and “Jugendleiter” (Eng. youth leader). Mismatches also happen when the longest string consists of an independently occurring word, e.g., “Kraftfahrtversicherung” (Eng. motor-vehicle insurance) and “Zusatzversicherung” (Eng. supplementary insurance) or the longest occurring word is an adjective, e.g., “Arabisch” (Eng. Arab) and “Arabische Liga” (Eng. Arab League). Both sources of mismatch, Levensthein distance and string match, could also occur in one single correspondence, e.g., “Leasinggesellschaft” (Eng. leasing company) and “Leistungsgesellschaft” (Eng. achieving society). Since the translations were equally used to build up correspondences they could also lead to a number of mismatches, e.g., “Brand” (Eng. incendiary) and “Marke” (Eng. brand). The same applies to indications of homonyms, e.g. “Samen (Volk)” (Eng. sami (people)) and “Volk” (Eng. people).

8.4 Conclusion

In this challenge, the overall improvement of the performance is encouraging. While it might not look impressive to beat simple baselines as ours at first sight, it is actually a notable achievement. The baselines are not only tailored for very high precision, benefitting from the fact that in many cases a consistent terminology is used, they also exploit additional knowledge about the labels. The matchers are general-purpose matchers that have to perform well in all challenges of the OAEI. Using the SKOS properties as annotation properties is a first step in order to make use of the many concept hierarchies provided on the Web.

In this regard, the improvement of F-measure for AML* is encouraging, since SKOS annotations may influence the matching result positively. The intellectual evaluation of new correspondences which have been created automatically has shown that matching tools are apparently still based exclusively on lexical approaches (comparison at string level). It becomes obvious that, instead, context knowledge is needed to avoid false correspondences. This context knowledge must clearly go beyond the mere consideration of translations and synonyms. One approach could be the consideration of the classification schemes of the Thesauri before establishing new correspondences. Taking into account the reference alignment, the highest confidence values should be assigned to the candidate correspondences that come from those classification schemes which have been most commonly mapped in the reference alignment.

9 Interactive matching

The interactive matching test case was evaluated at OAEI 2014 for the second time. The goal of this evaluation is to simulate interactive matching [26], in which a human expert is involved to validate correspondences found by the matching system. In the evaluation, we look at how user interaction may improve matching results.

For the evaluation, we use the conference data set (see 5) with the ra1 alignment, where there is quite a bit of room for improvement, with the best fully automatic, i.e., non-interactive matcher achieving an F-measure below 80%. The SEALS client was modified to allow interactive matchers to ask an oracle, which emulates a (perfect) user. The interactive matcher can present a correspondence to the oracle, which then tells the user whether the correspondence is right or wrong.

All matchers participating in the interactive test case support both interactive and non-interactive matching. This allows us to analyze how much benefit the interaction brings for the individual matchers.

9.1 Results

Overall, four matchers participated in the interactive matching track: AML, Hertuda, LogMap, and WeSeE-Match. The systems AML and LogMap have been further developed compared to last year, the other two ones are the same as last year. All of them implement interactive strategies that run entirely as a post-processing step to the automatic matching, i.e., take the alignment produced by the base matcher and try to refine it by selecting a suitable subset.

AML asks the oracle if the similarity variance between the matching algorithms AML employs is significant. Further, an alignment repair step is also performed interactively. Last year, AML presented all correspondences below a certain confidence threshold to the oracle, starting with the highest confidence values. LogMap checks all questionable correspondences using the oracle. Hertuda and WeSeE-Match try to adaptively set an optimal threshold for selecting correspondences. They perform a binary search in the space of possible thresholds, presenting a correspondence of average confidence to the oracle first. If the result is positive, the search is continued with a higher threshold, otherwise with a lower threshold.

Matcher	Precision		F-measure		Recall	
AML	**0.913	(0.85)	**0.801	(0.73)	**0.735	(0.64)
HerTUDA	0.790	(0.74)	0.582	(0.60)	0.497	(0.50)
LogMap	*0.888	(0.80)	*0.729	(0.68)	0.639	(0.59)
WeSeE	**0.734	(0.85)	0.473	(0.61)	0.404	(0.47)

Table 16. Results on the interactive matching task. The numbers in parantheses denote the results achieved without interaction. Significant differences between the interactive and non-interactive results are marked with * ($p < 0.05$) and ** ($p < 0.01$).

The results are depicted in Table 16. The largest improvement in F-measure, as well as the best overall result is achieved by AML, which increases its F-measure by seven percentage points (compared to the non-interactive results). Furthermore, AML shows a statistically significant increase in recall as well as precision, while all the other tools except for Hertuda show a significant increase in precision. The increase in precision is in all cases, except for AML, higher than the increase of recall. On the other hand, Hertuda, shows a decrease in recall, which cannot compensate for the increase in precision, and WeSeE shows a decrease in *both* recall and precision. Thus, we conclude that the interaction strategy used by those matchers is not as effective than those of the other participants.

When comparing to the results of last year [6], AML improved its F-measure by almost 10%. On the other hand, LogMap shows a slight decrease in recall, and hence, in F-measure. Compared to the results of the non-interactive conference track, the best interactive matcher (in terms of F-measure) is better than all non-interactive matching systems. Furthermore, the comparison to the non-interactive results show that there is a clear benefit of interactive matching – there, AML is also the best matching system, and still there is a significant improvement in both precision and recall when using interaction.

For further analyzing the effects of interaction and the efficiency at which the oracle is used, we also traced the number of interactions, both in absolute numbers and in relation to the size of the reference alignment. These measures are relevant in a practical setting, since the time of a domain expert validating is usually scarce, so an interactive matching tool should limit the number of interactions as much as possible. The results are depicted in Table 17.

It can be observed that LogMap has the lowest number of interactions with the oracle, while HerTUDA has the highest number, exposing roughly as many correspondences to the oracle as there are correspondences in the reference alignment. These observations show that, when comparing the tools, there is no clear trend showing that the number of interactions has a direct effect on the result quality – on the contrary, it is possible to build well performing tools using only few interactions.

Matcher	Total	Positive	Negative	Relative
AML	6.953	2.286	4.667	0.497
HerTUDA	12.285	1.952	10.333	0.996
LogMap	4.095	2.571	1.524	0.391
WeSeE	5.477	1.667	3.81	0.447

Table 17. Interactions of the individual matchers. The table depicts the average number of interactions used by the matchers (each interaction is the validation of one correspondence), the average number of positive and negative examples, and the relative number of interactions, i.e., divided by the size of the reference alignment.

Looking at the tools, it can be observed that current interactive matching tools mainly use interaction as a means to post-process an alignment found with fully automatic means. There are, however, other interactive approaches that can be thought of, which include interaction at an earlier stage of the process, e.g., using interaction for parameter tuning [28], or determining anchor elements for structure-based matching approaches using interactive methods. The maximum F-measure of 0.801 achieved shows that there is still room for improvement. Furthermore, different variations of the evaluation method can be thought of, including different noise levels in the oracle's responses, i.e., simulating errors made by the human expert, or allowing other means of interactions than the validation of single correspondences, e.g., providing a random positive example, or providing the corresponding element in one ontology, given an element of the other one.

10 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 [18] 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) [31, 32].

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 [23]. 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. OA4QA track [33], introduced in 2014, evaluates those 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., Optique project[15] [21, 31]. The integration via ontology alignment is required since only the vocabulary of the DB-Ontology is connected to the data. The *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.

10.1 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.

[15] http://www.optique-project.eu/
[16] http://dblp.uni-trier.de/xml/

10.2 Query Evaluation Engine

The evaluation engine considered is an extension of the OWL 2 reasoner HermiT, known as OWL-BGP[17] [22]. OWL-BGP is able to process SPARQL queries in the SPARQL-OWL fragment, under the OWL 2 Direct Semantics entailment regime [22]. 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 support queries where variables occur within complex class expressions or bind to class or property names.

10.3 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.

10.4 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 [31]. 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 corre-

[17] https://code.google.com/p/owl-bgp/

spondences from the reference alignment.

$$confof{:}Scholar \equiv ekaw{:}Student \quad (1)$$

$$confof{:}Scholar \sqsubseteq confof{:}Participant \quad (2)$$

$$confof{:}Participant \equiv ekaw{:}Conf\_Participant \quad (3)$$

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 mapping removal, in this case the removal of either Axiom 1, or Axiom 3 could be possible repairs. Finally, for stategies including the input ontologies as a possible repair target, the removal of Axiom 2 can be proposed as a legal solution to the problem.

10.5 Results

Table 18 shows the average precision, recall and f-measure results for the whole set of queries: AML, LogMap, LogMap-C and XMap were the only matchers whose alignments allowed to answer all the queries of the evaluation.

LogMap was the best performing tool for what concerns averaged precision, recall and f-measure, closely followed by LogMap-C and AML. XMap, despite being able to produce an alignment not leading to unsatifiability during query answering, did not perform as well.

Considering Table 18, the difference in results between the publicly available reference alignment of the *Conference* track (*ra1*) and its repaired version (*rar1*) was not significant and, as expected, affected precision. Most of the differences between *ra1* and *rar1* are related to conservativity violations, and this is reflected by a reduced precision employing *rar1* w.r.t. *ra1*. However, the f-measure ranking between the two reference alignments is (mostly) preserved. If we compare Table 18 (the results of the present track) and Table 7 (the results of *Conference* track) we can see that the top-4 matcher ranking coincides, even if with a slight variation. But, considering *rar1* alignment, the gap between the top-4 matcher 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 *OA4QA* track evaluation.

Comparing Table 18 and Table 19 (measuring the degree of incoherence of the computed alignments of the *Conference* track) it seems that a negative correlation between the ability of answering queries and the average degree of incoherence of the matchers do exists. For instance, taking into account the different positions in the ranking of AOT, we can see that logical violations are definitely penalized more in our test case than in the traditional *Conference* track, due to its target scenario. MaasMatch, 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 (5 out of 18).

LogMap-C, 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 to one of the queries (the result set was empty as an effect of correspondence removal).

Table 18. 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.
LogMap	18/18	0.750	0.750	0.750	0.729	0.739	0.750
AML	18/18	0.722	0.708	0.694	0.701	0.697	0.694
LogMap-C	18/18	0.722	0.708	0.694	0.722	0.708	0.694
XMap	18/18	0.556	0.519	0.487	0.554	0.518	0.487
RSDLWB	15/18	0.464	0.471	0.479	0.407	0.431	0.458
OMReasoner	15/18	0.409	0.432	0.458	0.407	0.431	0.458
LogMapLite	11/18	0.409	0.416	0.423	0.351	0.375	0.402
MaasMatch	5/18	0.223	0.247	0.278	0.203	0.235	0.278
AOTL	6/18	0.056	0.056	0.056	0.056	0.056	0.056
AOT	0/18	0.000	0.000	0.000	0.000	0.000	0.000

Table 19. Incoherences in the alignment computed by the participants to the Conference track. The values in the "Alignment size" and "Inc. Degre" columns represent averages over the 21 computed alignments.

Matcher	Alignment size	Inc. alignments	Inc. Degree
AML	10.95	0/21	0%
AOT	59.17	18/21	40.4%
AOTL	14.67	17/21	15.1%
LogMap	10.71	0/21	0%
LogMap-C	10.24	0/21	0%
LogMapLite	9.91	7/21	5.4%
MaasMatch	33.00	19/21	21%
OMReasoner	8.10	4/21	2.5%
RSDLWB	8.33	4/21	2.5%
XMap	8.14	0/21	0%

10.6 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 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 [27] 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 *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.

11 Instance matching

The instance matching track evaluates the performance of matching tools when the goal is to detect the degree of similarity between pairs of items/instances expressed in the form of OWL Aboxes. The track is organized in two independent tasks, namely the *identity recognition task* (*id-rec task*) and the *similarity recognition task* (*sim-rec task*).

In both tasks, participants received two datasets called source and target, respectively. The datasets contain instances describing famous books with different genres and topics. We asked the participants to discover the matching pairs, i.e., links or mappings, among the instances in the source dataset and the instances in the target dataset. Both tasks are blind, meaning that the set of expected mappings, i.e., reference link-set, is not known in advance by the participants.

11.1 Results of the identity recognition task

The id-rec task is a typical evaluation task of instance matching tools where the goal is to determine when two OWL instances describe the same real-world entity. The datasets of the id-rec task have been produced by altering a set of original data with the aim to generate multiple descriptions of the same real-world entities where different languages and representation formats are employed. We stress that an instance in the source dataset can have none, one, or more than one matching counterparts in the target dataset. The source dataset is an Abox containing 1330 instances described through 4 classes, 5 datatype properties, and 1 annotation property. The target dataset contains 2649 instances described through 4 classes, 4 datatype properties, 1 object property, and 1 annotation property.

We asked the participants to match the instances of the class `http://www.instancematching.org/ontologies/oaei2014#Book` 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 real-world entity.

The participants to the identity recognition task are InsMT, InsMTL, LogMap, LogMap-C, and RiMOM-IM. 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 all the instances in the target dataset that has been generated as an altered description of i_1.

The evaluation has been performed by calculating precision, recall, and F-measure and results are provided in Figure 3.

	Precision	F-measure	Recall
InsMT	0.00	0.00	0.78
InsMTL	0.00	0.00	0.78
LogMap	0.60	0.10	0.05
LogMap-C	0.64	0.08	0.04
RiMOM-IM	0.65	0.56	0.49

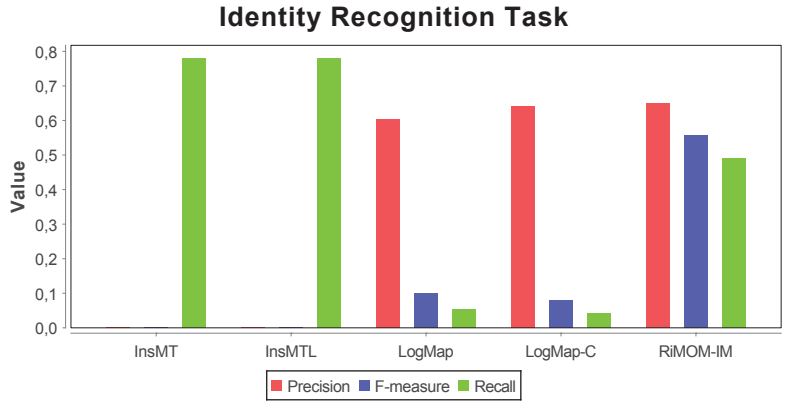


Fig. 3. Results of the id-rec task

A first comment on the id-rec results is that the quality of the alignment is in general not very high, especially concerning the recall. Basically, the main kind of transformation that we performed is to transform the structured information into an unstructured version of the same information. As an example, for many instances we substitute labels and book titles with a set of keywords taken from the instance description. The result of this kind of transformation is that we have a second instance where it is possible to retrieve the same terms appearing in the label and titles but with no reference to the corresponding metadata. Moreover, a further challenge was the substitution of the original English terms with the corresponding Italian translation. We empirically proved that human users are able to capture the correct links also in case of these transformations, but automatic tools still have problems in several cases. We also note a very different behavior of RiMOM-IM and LogMap/LogMap-C with respect to InsMT/InsMTL. The former two tools produce links that are quite often correct (resulting in a good precision) but they fail in capturing a large number of the expected links (resulting in a low recall), especially in the case of LogMap/LogMap-C. Instead, InsMT/InsMTL have the opposite behavior. This is due to the fact that InsMT/InsMTL produces a large number of links having more or less the same similarity value. This means that the probability of capturing a correct link is high, but the probability of a retrieved link to be correct is low, resulting then in a high recall, but a very low precision.

11.2 Results of the similarity recognition task

The sim-rec task focuses on the evaluation of the similarity degree between two OWL instances, even when the two instances describe different real-world entities. Similarity recognition is new in the instance matching track of OAEI, but this kind of task is becoming a common issue in modern web applications where large quantities of data are daily published and usually need to be classified for effective fruition by the final user.

The datasets of the sim-rec task have been produced through crowdsourcing by employing the Argo system[18] [5]. More than 250 workers have been involved in the crowdsourcing process to evaluate the degree of similarity between pairs of instances describing books. Crowdsourcing activities have been organized into a set of HITs (Human Intelligent Task) assigned to workers for execution. A HIT is a question where the worker is asked to evaluate the degree of similarity of two given instances. The worker exploits the instances, i.e., book descriptions, "at a glance" and she/he has to specify her/his own perceived similarity by assigning a degree in the range [0,1].

We asked the participants to match the instances of the class `http://www.instancematching.org/ontologies/oaei2014#Book` in the source dataset against the instances of the corresponding class in the target dataset. we asked to produce a complete set of links/mappings between any pair of instances. The source dataset contains 173 book instances and the target dataset contains 172 book instances, then we expected to receive a set of $173 * 172 = 29756$ links as a result, each one featured by a degree of similarity in the range $[0, 1]$.

The participants to the similarity recognition task are InsMT and RiMOM-IM. For evaluation, we call *reference alignment* the link-set obtained through crowdsourcing, where each link $l_c(i_1, i_2, \sigma^c_{12})$ denotes that workers assigned a similarity degree σ^c_{ij} to the pair of instances i_1 and i_2. The cardinality of the reference alignment is 4104 links. In the analysis, we are interested in comparing the similarity degree σ^c of a link l_c against the similarity degree σ^i and σ^r calculated by InsMT and RiMOM-IM, respectively (see Figure 4). The goal of this comparison is to analyze how different is the human perception of similarity with respect to the automatic matching tools.

In the diagram, for a link $l_c(i_1, i_2, \sigma^c_{12})$, we plot i) a red line to represent the gap between the similarity degree of the reference link-set and and the corresponding value calculated by InsMT (i.e., $\sigma^c_{12} - \sigma^i_{12}$), and ii) a blue line to represent the gap between the similarity degree of the reference alignment and the corresponding value calculated by RiMOM-IM (i.e., $\sigma^c_{12} - \sigma^r_{12}$). For the sake of readability, the links of the reference links are sorted according to the associated similarity degree. Moreover, a black line is the marker used for 0-values, i.e., the minimum gap between the reference links and the tools result. When the reference link similarity (i.e., the similarity as it is perceived by human workers in the crowd) is higher than the similarity degree calculated by the participating tool, the value of the gap between the two is positive, meaning that the tool underestimated the similarity of a pair of instances in the two datasets with respect to the human judgment. On the contrary, when the tool reference link similarity is lower than the tool resulting value, the gap between the two values is negative, meaning that

[18] `http://island.ricerca.di.unimi.it/projects/argo/`

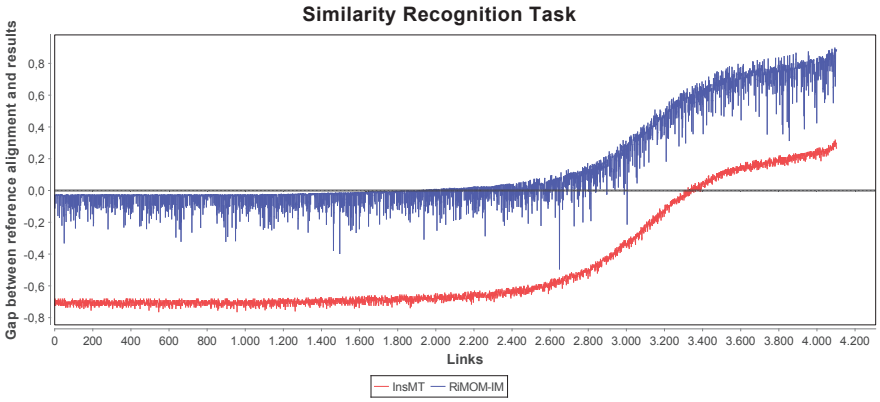


Fig. 4. Results of the sim-rec task: gap between the similarity degrees calculated by InsMT and RiMOM-IM and the reference alignment

the tool overestimated the similarity between the instances with respect to the human judgment. By analyzing Figure 4, we note that InsMT produces homogeneous similarity values for the links, resulting in a more homogeneous distribution of the similarity degrees. However, the average gap value from the expected degrees of similarity is quite high and the number of similarity degrees that have been overestimated (resulting in a negative gap) is high as well. On the contrary, for RiMOM-IM, we have higher variability in the similarity degrees but a large number of links have a similarity degree very near to the expected value. Moreover, in case of RiMOM-IM, the number of overestimated similarity values is more or less the same than the number of underestimated values. Furthermore, the gap between the results of the two tools and the expected links has been measured by the Euclidean distance considering each link as a dimension, in order to compare the similarity of the same correspondence. As a result, we have $d(\text{InsMT}) = 37.03$ and $d(\text{RiMOM-IM}) = 21.83$.

As a further evaluation analysis, we split the range $[0, 1]$ of possible similarity degrees into ten smaller ranges of size 0.1 that we call *range-of-gap*. A range-of-gap rd is populated with those links whose gap from the reference alignment is in the range of rd. Consider a link $l_c(i_1, i_2, \sigma_{12}^c)$. For InsMT and RiMOM-IM, the link l_c is placed in the range-of-gap corresponding to the value $\mid \sigma_{12}^c - \sigma_{12}^i \mid$ and $\mid \sigma_{12}^c - \sigma_{12}^r \mid$, respectively. The results of the analysis by range-of-gap are provided in Figure 5. From the bar chart of Figure 5, we note that the results of RiMOM-IM are better of InsMT. In fact, RiMOM-IM is capable of retrieving a correct degree of similarity, i.e., with a difference from the expected value lower than 0.1, for about 2400 links of the 4104 in the reference alignment ($\approx 60\%$). This result can be considered as a very good performance and shows how RiMOM-IM is capable of adequately simulating the human behavior in the evaluation of the similarity between two real object descriptions. In case of InsMT, the peculiar behavior of the tool is to produce the largest part of the similarity values in the small range [0.6, 0.8]. As a consequence, the majority of the links are in the range-of-gap [0.6-0.7] and [0.6-0.8], which denotes a remarkable difference between the automatic result and the human judgment.

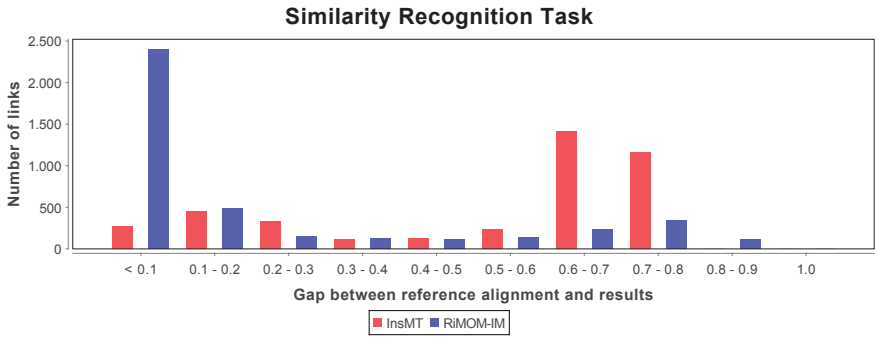


Fig. 5. Results of the sim-rec task: analysis by range-of-gap

12 Lesson learned and suggestions

Here are lessons learned from running OAEI 2014:

A) This year indicated again that requiring participants to implement a minimal interface was not a strong obstacle to participation. Moreover, the community seems to get used to the SEALS infrastructure introduced for OAEI 2011.
B) As already proposed last year, 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.
C) Now that all tools are run in exactly the same configuration across all test cases, some discrepancies appear across such cases. For instance, benchmarks expect only class correspondences in the name space of the ontologies, some other cases expect something else. This is a problem, which could be solved either by passing parameters to the SEALS client (this would make its implementation heavier), by specifying a flag in test descriptions that can be tested by matcher interfaces, or by post processing results (which may be criticized).
D) In the OAEI 2013, [27] raised and documented objections (on validity and fairness) to the way reference alignments are made coherent with alignment repair techniques. This year we created a new reference alignment in the largebio track that mitigates this issue.
E) Last years we reported that we had many new participants. This year we got new participants as well, however the overall participation has decreased.
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.
G) Last year we proposed to include provenance information in reference alignments. We did not achieved this goal mostly due to the heaviness of the prov-o ontology. This is, anyway, a goal worth pursuing.
H) The SEALS repositories are still hosted by STI because moving them to Madrid revealed more difficult than expected. A solution has to be found for this transfer.

13 Conclusions

OAEI 2014 saw a decreased number of participants. We hope to see a different 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, for the first year, all systems participating in the anatomy track finished in less than 1 hour. As usual, most of the systems favour 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.

A novelty of this year was the evaluation of ontology alignment 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.

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 [30]. 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 STI Innsbruck for providing the necessary infrastructure to maintain 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 help with incoherence evaluation within the conference and his support of the anatomy 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).

Bernardo Cuenca Grau, 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 and Bernardo have 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.

Cássia Trojahn dos Santos and Roger Granada are also partially supported by the CAPES-COFECUB Cameleon project number 707/11.

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/ 2014).

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. Silvana Castano, Lorenzo Genta, and Stefano Montanelli. Leveraging Crowdsourced Knowledge for Web Data Clouds Empowerment. In *Proc. of the 7th IEEE Int. Conference on Research Challenges in Information Science (RCIS 2013)*, Paris, France, 2013.
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. Jérôme David, Jérôme Euzenat, François Scharffe, and Cássia Trojahn dos Santos. The alignment API 4.0. *Semantic web journal*, 2(1):3–10, 2011.
8. 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.
9. 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.
10. 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.
11. 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.
12. 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.
13. 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.
14. Jérôme Euzenat, Maria Rosoiu, and Cássia Trojahn dos Santos. Ontology matching benchmarks: generation, stability, and discriminability. *Journal of web semantics*, 21:30–48, 2013.
15. Jérôme Euzenat and Pavel Shvaiko. *Ontology matching*. Springer-Verlag, Heidelberg (DE), 2nd edition, 2013.
16. 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.
17. 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.
18. 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.
19. 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.
20. 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.
21. Evgeny Kharlamov, Martin Giese, Ernesto Jiménez-Ruiz, Martin G. Skjæveland, Ahmet Soylu, Dmitriy Zheleznyakov, Timea Bagosi, Marco Console, Peter Haase, Ian Horrocks, Sarunas Marciuska, Christoph Pinkel, Mariano Rodriguez-Muro, Marco Ruzzi, Valerio

Santarelli, Domenico Fabio Savo, Kunal Sengupta, Michael Schmidt, Evgenij Thorstensen, Johannes Trame, and Arild Waaler. Optique 1.0: Semantic access to big data: The case of norwegian petroleum directorate's factpages. In *Proceedings of the ISWC 2013 Posters & Demonstrations Track*, pages 65–68, 2013.
22. 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.
23. Christian Meilicke. *Alignment Incoherence in Ontology Matching*. PhD thesis, University Mannheim, 2011.
24. 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.
25. Boris Motik, Rob Shearer, and Ian Horrocks. Hypertableau reasoning for description logics. *Journal of Artificial Intelligence Research*, 36:165–228, 2009.
26. 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.
27. 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.
28. Dominique Ritze and Heiko Paulheim. Towards an automatic parameterization of ontology matching tools based on example mappings. In *Proc. 6th ISWC ontology matching workshop (OM), Bonn (DE)*, pages 37–48, 2011.
29. Emanuel Santos, Daniel Faria, Catia Pesquita, and Francisco Couto. Ontology alignment repair through modularization and confidence-based heuristics. *CoRR*, abs/1307.5322, 2013.
30. 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.
31. 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.
32. 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.
33. 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.
34. York Sure, Oscar Corcho, Jérôme Euzenat, and Todd Hughes, editors. *Proc. ISWC Workshop on Evaluation of Ontology-based Tools (EON), Hiroshima (JP)*, 2004.

Linköping, Mannheim, Grenoble, Milano, Lisbon, Oxford, Porto Alegre, Toulouse,
Köln, Trento, Genova, Prague
October 2014

AgreementMakerLight Results for OAEI 2014

Daniel Faria[1], Catarina Martins[2,3], Amruta Nanavaty[4], Aynaz Taheri[4], Catia Pesquita[1,2], Emanuel Santos[2], Isabel F. Cruz[4], and Francisco M. Couto[1,2]

[1] LASIGE, Faculdade de Ciências, Universidade de Lisboa, Portugal
[2] Departamento de Informática, Faculdade de Ciências, Universidade de Lisboa, Portugal
[3] INESC-ID, Universidade de Lisboa, Portugal
[4] ADVIS Lab, Department of Computer Science, University of Illinois at Chicago, USA

Abstract. AgreementMakerLight (AML) is an automated ontology matching framework based on element-level matching and the use of external resources as background knowledge. This paper describes the configuration of AML for the OAEI 2014 competition and discusses its results.
Our goal this year was broadening the scope of AML by delving into aspects such as translation and structural matching, while reinforcing the key aspects behind its success last year (i.e., element-level matching, the use of background knowledge, and alignment repair).
AML's participation in the OAEI 2014 was very successful, as it obtained the highest F-measure in 6 of the 8 ontology matching tracks.

1 Presentation of the system

1.1 State, purpose, general statement

AgreementMakerLight (AML) is an automated ontology matching system derived from AgreementMaker [1, 2] and developed to handle large ontology matching problems. It combines the design principles of AgreementMaker (flexibility and extensibility) with a strong focus on efficiency [6]. Furthermore, it draws on the knowledge accumulated in AgreementMaker by reusing and adapting several of its components, but also includes a growing number of novel components.
AML is primarily based on lexical matching techniques, with an emphasis on the use of external resources as background knowledge. It also emphasizes alignment coherence, featuring an improved alignment repair module.
While initially AML was mainly focused on the biomedical domain, we have striven to expand its scope throughout the last year, and it is now a general-purpose ontology matching system. We have also moved towards full automation by employing a general-purpose core matching strategy complemented with an automated background knowledge selection algorithm [5].

1.2 Specific techniques used

The AML workflow for the OAEI 2014 comprises nine different steps, as 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. The key differences from last year's workflow are the introduction of the translation and structural matching steps.

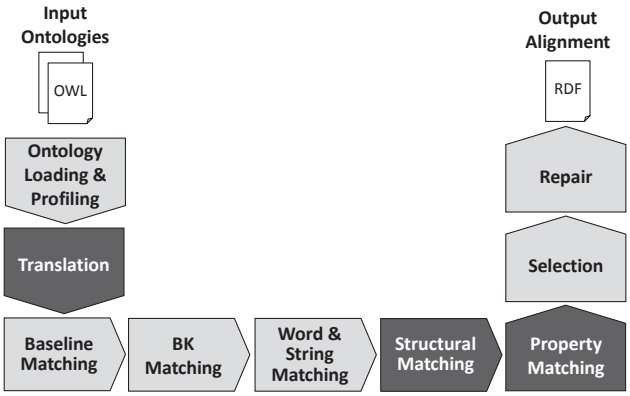


Fig. 1. The AgreementMakerLight matching workflow for the OAEI 2014. Steps in dark gray are conditional.

Ontology Loading & Profiling AML employs the OWL API [7] to read the input ontologies then retrieve the necessary information to populate its own data structures [6]:

- Class localNames, labels and synonym annotations 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 [12], and by removing name sections within parenthesis.
- Property names, types, domains, and ranges are stored in the *PropertyList* of the corresponding ontology.
- Relations between classes (including disjointness) and between properties are stored in a global *RelationshipMap*.

Note that AML currently 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 the property/class ratio.

Translation AML features an automatic translation module based on Microsoft® Translator. 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 those of the second and vice-versa. 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.

Baseline Matching AML employs an efficient weighted string-equivalence algorithm, the *Lexical Matcher* [6], to obtain a baseline class alignment between the input ontologies. The *Lexical Matcher* has been updated to handle multi-language ontologies, by matching only class names in the same language.

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) [9], the Human Disease Ontology (DOID) [14], the Medical Subject Headings (MeSH) [10], and the WordNet [8].
The WordNet is only used for small English language ontologies, as it is prone to produce erroneous mappings in other settings. It is used through the JAWS API [1] and with the *Lexical Matcher*. The remaining three background knowledge sources are tested in all non-small single-language problems, by measuring their mapping gain over the baseline alignment [5]. When their mapping gain is very high ($\geq$20%), they are used to extend the *Lexicons* of the input ontologies [12]; otherwise, when it is above the minimum threshold (2%) their alignment is merged with 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 the MeSH 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*) [6]. 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-intensive.
For small ontologies, AML also employs the new *Multi-Word Matcher*, which matches closely related multi-word names that have matching words and/or words with common WordNet synonyms or close hypernyms.

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 [4]. 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.

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

Selection AML employs a greedy selection algorithm, the *Ranked Selector* [6], 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 the interactive matching track, AML employs an interactive selection algorithm which asks the user for feedback about mappings which are below a high similarity threshold (70%) and have a significant variance (with regard to similarity) between matching algorithms. The algorithm stops when a given threshold of negative answers is reached. This algorithm is based on AgreementMaker's user feedback module [3].

Repair AML employs a heuristic repair algorithm to ensure that the final alignment is coherent [13].
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.

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).

1.5 Link to the set of provided alignments

The alignments generated by AML for the OAEI 2014 are available at the SOMER project page (http://somer.fc.ul.pt/).

2 Results

2.1 Anatomy

AML had the highest F-measure and recall this year (and all time) in this track, registering a slight improvement over last year's result (of 0.2%). This improvement was

Table 1. AgreementMakerLight global results on all the OAEI 2014 tracks.

Track	Precision	Recall	F-Measure	Run Time
Anatomy	95.6%	93.2%	94.4%	28 sec
Interactive Matching	91.3%	73.5%	80.1%	19 sec
Benchmark				
biblio	92%	39%	55%	49 sec
cose	46%	46%	46%	140 sec
dog	98%	58%	73%	1506 sec
Conference				
Reference 1	85%	64%	73%	14 sec
Reference 2	80%	58%	67%	
Large Biomedical Ontologies				
Average	90.6%	75.2%	81.9%	307 sec
FMA-NCI small	96.0%	89.9%	92.8%	27 sec
FMA-SNOMED small	92.6%	74.2%	82.4%	126 sec
SNOMED-NCI small	91.7%	72.4%	80.9%	831 sec
FMA-NCI whole	83.2%	85.6%	84.4%	112 sec
FMA-SNOMED whole	89.1%	64.7%	74.9%	251 sec
SNOMED-NCI whole	91.2%	64.5%	75.6%	497 sec
Library				
New OWL	82.4%	77.8%	80.0%	68 sec
Old OWL	71.6%	74.8%	73.1%	71 sec
Multifarm				
Different Ontologies	57%	53%	54%	8 min
Same Ontologies	95%	48%	62%	
Ontology Alignment for Query Answering				
Original Reference	72.2%	69.4%	70.4%	N/A
Repaired Reference	70.1%	69.4%	69.1%	

mainly due to the addition of MeSH as a background knowledge source.

2.2 Benchmark

AML had a good performance in *dog*, ranking third in F-measure, but average performances in the other two test suites. This difference is due to the fact that AML does not handle ontology instances, which are present in the other two test suites, but not in *dog*.

2.3 Conference

AML ranked first in F-measure and recall, and second in precision this year, with a considerable improvement (3% F-measure) over last year's results. This improvement is due to the refinements in the Word and String Matching step.

2.4 Interactive Matching

AML ranked first in F-measure, recall, and precision this year, with a considerable improvement (7.2% F-measure) over last year's result. This improvement is partially due to the (non-interactive) refinements in the Word and String Matching step, but mainly due to the refinement of the interactive selection algorithm. The latter is evidenced by the fact that the difference between AML's interactive and non-interactive performance increased since last year (from 3 to 7.1% in F-measure) while the number of user interactions was approximately the same.

2.5 Large Biomedical Ontologies

AML had the highest F-measure in all six tasks this year, and the highest F-measure of all time in four of them. It improved substantially in all tasks, thanks to the addition of new background knowledge sources (MeSH and DOID) and the refined selection step.

2.6 Library

AML ranked first in F-measure, precision and recall in this track, having the highest F-measure of all time when using the new OWL conversion of the Library thesauri. AML's F-measure using the old OWL conversion was approximately the same as last year, but it had a higher precision and a lower recall due to a more stringent selection step. The improvement when using the new OWL conversion is due to the conversion's differentiation of *skos:altLabel* and *skos:prefLabel*. This effectively enables AML's lexical weighting scheme, which greatly improves its ability to score and select mappings.

2.7 Multifarm

AML had the highest F-measure and recall in both modalities of this track this year (and the highest F-measure of all time) and also the highest precision in matching same ontologies. These results show that, while simple, AML's new translation module is effective. The improvement over last year was dramatic, as last year AML did not perform translation.

2.8 Ontology Alignment for Query Answering

AML ranked second and third in F-measure in this new track, when evaluated on the original and repaired reference alignment respectively. While good, these results do not reflect the fact that AML produced the best set of alignments for the Conference ontologies used in this track, as the number of queries performed was too small to be representative.

3 General comments

3.1 Comments on the results

AML was very successful in this year's evaluation, ranking first in F-measure in 6 of the 8 ontology matching tasks. Furthermore, AML improved substantially over last year's evaluation, overcoming several of its limitations. Throughout the past year, we have striven to make AML a more complete ontology matching system while maintaining a strong emphasis on efficiency, which is accurately depicted in the results.

3.2 Discussions on the way to improve the proposed system

The one key feature still missing from AML is handling (and matching) ontology instances, so this is the aspect where it could improve the most. We are also interested in enabling AML to read SKOS thesauri, to broaden its applicability.

3.3 Comments on the OAEI test cases

This year's Large Biomedical Ontologies reference alignments marks a significant improvement over previous years, as the use of a 'soft' repair (where mappings are flagged rather than removed) makes the evaluation less biased [11].
Currently, our main concern about the evaluation is the (in)completeness of some of the reference alignments, particularly in the Large Biomedical Ontologies and (to a lesser degree) Anatomy tracks. Upon analyzing the alignment produced by AML for these tracks, we observed that many of the false positives were in fact correct mappings that were absent from the reference alignment. Incomplete reference alignments undermine OAEI's evaluation effort, so albeit cumbersome, completing them is paramount. On our part, we will share with the track organizers the false positive mappings found by AML that we deem to be true positives, upon a more extensive analysis.
We would also like to comment on the fact that the Conference track's reference alignment 1 includes many mappings that are apparently erroneous (as they were removed upon creating reference alignment 2). This is evidenced by the considerable drop in precision between references 1 and 2 observed for all systems. While we see the merit in a blind evaluation, we would expect that, if a partial reference alignment is provided to systems, it be fully correct. This is especially relevant given that the reference alignment 1 is used in several other OAEI tracks.
Finally, while we recognize the importance of evaluating ontology alignment applications, as per the new Query Answering track, we hope that in subsequent OAEI editions this evaluation be more representative of the underlying alignments.

4 Conclusion

AML's OAEI 2014 participation was a success, as it ranked first in F-measure in 6 of the 8 ontology matching tracks. This success reflects the effort put into the development of AML throughout the last year, which focused on increasing efficiency and automation, and particularly on expanding AML's scope.

Acknowledgments

DF, CP, ES, CM and FMC were funded by the Portuguese FCT through the SOMER project (PTDC/EIA-EIA/119119/2010) and the LASIGE Strategic Project (PEst-OE/EEI/UI0408/2014). The research of IFC, AN and AT was partially supported by NSF Awards CCF-1331800, IIS-1213013, IIS-1143926, and IIS-0812258 and by a UIC-IPCE Civic Engagement Research Fund Award.
We would like to thank Pedro do Vale, Joana Pinto, and Cláudia Duarte for their collaboration in developing AML.

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, C. Stroe, and M. Palmonari. Interactive user feedback in ontology matching using signature vectors. *2012 IEEE 30th International Conference on Data Engineering*, 0:1321–1324, 2012.
4. I. F. Cruz and W. Sunna. Structural alignment methods with applications to geospatial ontologies. *Transactions in GIS*, 12(6):683–711, 2008.
5. D. Faria, C. Pesquita, E. Santos, I. F. Cruz, and F. M. Couto. Automatic Background Knowledge Selection for Matching Biomedical Ontologies. *PLoS One*, In Press, 2014.
6. 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.
7. M. Horridge and S. Bechhofer. The owl api: A java api for owl ontologies. *Semantic Web*, 2(1):11–21, 2011.
8. G. A. Miller. WordNet: A Lexical Database for English. *Communications of the ACM*, 38(11):39–41, 1995.
9. 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.
10. 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.
11. C. Pesquita, D. Faria, E. Santos, and F. M. Couto. To repair or not to repair: reconciling correctness and coherence in ontology reference alignments. In *ISWC International Workshop on Ontology Matching (OM)*, CEUR Workshop Proceedings, 2013.
12. 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.
13. E. Santos, D. Faria, C. Pesquita, and F. M. Couto. Ontology alignment repair through modularization and confidence-based heuristics. arXiv:1307.5322, 2013.
14. 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.

AOT / AOTL Results for OAEI 2014

Abderrahmane Khiat[1], Moussa Benaissa[1]

[1] LITIO Lab, University of Oran, BP 1524 El-Mnaouar Oran, Algeria
abderrahmane_khiat@yahoo.com
moussabenaissa@yahoo.fr

Abstract. In this paper, we describe our two ontology alignment systems AOT and AOTL respectively. The AOT system uses different terminological matchers with a local filter and the AOTL system consists in combining the different terminological with linguistic matchers. The AOT and AOTL systems are designed for the ontology matching tracks in order to discover new semantic correspondences between entities of different ontologies to be aligned. This is the first participation of AOT and AOTL at OAEI 2014, we present the results obtained by running the first version of our systems in different tracks of OAEI 2014 evaluation campaign.

1 Presentation of the system

1.1 State, purpose, general statement

AOT (Ontology Alignment at Terminological level) and AOTL (Ontology Alignment at Terminological and Linguistic level) are automatic ontology alignment systems realized for the purpose to solve the problem of ontology Matching. The AOT system uses various terminological matchers with a local filter in order to find correspondences between ontologies to be aligned. Contrary to AOT system, AOTL combines the similarities calculated by the various string matching algorithms at terminological level without a local filter, then these similarities are combined with similarities calculated using an external resource WordNet i.e. at linguistic level. The next step (for AOTL system) consists in combining the similarities by gives the priority to linguistic matcher; otherwise we have used an average aggregation method. Finally AOT and AOTL applied a filter in order to identify the alignment

For AOT system, we have proposed a local filter (section 1.2.1.3) in order to select better correspondences and we envision to use AOT in order to study the system behavior using different aggregation and filter methods and proposing in the future more efficient filters and aggregation methods and add other matchers.

For AOTL system, we have used an external resource WordNet in order to select semantic correspondences and we plan to use AOTL in order to discover new

semantic correspondences more than select the better one i.e. we are interested in recall more than precision, of course with good the f-measure (balanced).

The details of each step of our systems are described in the following section.

1.2 Specific techniques used

The process of AOT and AOTL systems consists in the following two successive steps: 1) Calculation of Similarities and 2) Combination and Extraction of Alignment.

A. AOT system

1.2.1 Step 1: Calculation of Similarities

1.2.1.1 Phase 1: Extraction of Entities of the Ontologies

In this phase, our system takes as input the two ontologies to be aligned and extract their entities: names, labels, properties (data property and object property) and without forgetting the instances.

1.2.1.2 Phase 2: The Applied Matchers

In this phase, our system calculates the similarities between entities extracted in previous phase, using various string-based matching algorithms. More precisely the different string-based matching algorithms used are: levenshtein-distance, block-distance, Jaro, SLIM-Winkler, Jaro-Winkler, Smith-Waterman and Needleman-Wunsch. The calculations of similarities by each string matching algorithm are represented in matrix.

1.2.1.3 Phase 3: Local Filter

In this phase, 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.

1.2.2 Step 2: Combination and Extraction of Alignment

1.2.2.1 Phase 1: Aggregation of Similarities

In this phase, 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.

1.2.2.2 Phase 2: Global Filter and Identification of Alignment

In this final phase, our system applies a second filter on the matrix combined (result of the previous step) in order to select the correspondences found using the maximum strategy with a threshold.

B. AOTL system

We mention in this section the difference between AOT and AOTL system.

First, we have added another matcher at linguistic level for AOTL system in second phase "The applied Matchers", we have used an external dictionary WordNet.

AOTL does not use phase "Local Filter", the similarities calculated by each matcher are represented in matrix without a local filter.

In the phase "Aggregation of Similarities", AOTL system gives priority to WordNet i.e. if the similarity value calculated using WordNet is greater than the similarity value calculated using string matching algorithms, the similarity value of the matrix combined is equal to the similarity calculated using WordNet, else we use the average aggregation method. The result of the aggregation is represented in a matrix.

1.3 Adaptations made for the evaluation

We do not have made any specific adaptation for the first version of AOT and AOTL, for OAEI 2014 evaluation campaign. All parameters are the same for different tracks of OAEI 2014.

1.4 Link to the system and parameters file

The first version of AOT and AOTL systems submitted to OAEI 2014 can be downloaded from seal-project at http://www.seals-project.eu/.

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

The results of AOT and AOTL systems can be downloaded from seal-project at http://www.seals-project.eu/.

2 Results

In this section, we present the results obtained by running AOT and AOTL on different tracks of OAEI 2014 evaluation campaign i.e. on the tracks: Benchmark, Conference, Multifarm and Anatomy.

2.1 Benchmark

The Benchmark track contains different series which contain reference ontologies of different sizes and from different domains. The AOT system uses various string-based matching algorithms in order to find correspondences between entities of the two ontologies to be aligned and the AOTL system use another matcher at linguistic level in order to select semantic correspondences. However when these ontologies do not contain terminological information (limited information or random strings) our systems fails to identify the alignment.

The table 1 below presents the results obtained by running AOT and AOTL on the Benchmark tracks of OAEI 2014 evaluation campaign i.e. H-mean of our systems on tracks: biblio and finance.

System	Test group	H-mean Prec.	H-mean Rec	H-mean f-Measure
AOT	Biblio	0.96	0.50	0.68
	Finance	0.77	0.65	0.70
AOTL	Biblio	0.85	0.67	0.75
	Finance	0.75	0.63	0.68

Table 1. The results of AOT and OATL on the Benchmark track of OAEI 2014.

2.2 Anatomy

The Anatomy track contains two large ontologies that describe the biomedical domain of human and mouse anatomy. The Table 2 shows the results obtained by running AOT and AOTL on Anatomy track of OAEI campaign 2014.

System	Test	H-mean Prec.	H-mean Rec	H-mean f-Measure
AOT	Anatomy	0.436	0.775	0.558
AOTL	Anatomy	0.707	0.078	0.14

Table 2. The results of AOT and OATL on the Anatomy track of OAEI 2014.

2.3 Conference

The conference track contains about 16 ontologies that describe the same domain (conference organization). The Table 3 presents the results obtained by running AOT and AOTL on Conference track of OAEI campaign 2014.

System	Test	H-mean Prec.	H-mean Rec	H-mean f-Measure
AOT	Conference	0.8	0.48	0.59
AOTL	Conference	0,78	0,42	0,55

Table 3. The results of AOT and OATL on Conference track of OAEI 2014.

2.4 Multifarm

The Multifarm track contains different ontologies translated into different languages. Our systems AOT and AOTL do not deal efficiently (for now) with the Multifarm track. The Table 4 presents the results obtained by running AOT and OATL on Multifarm track of OAEI campaign 2014.

System	Test	H-mean Prec.	H-mean Rec	H-mean f-Measure
AOT	Diff-ontologies	0,02	0,17	0,04
	Same-ontologies	0,11	0,12	0,12
AOTL	Diff-ontologies	0,10	0,2	0,3
	Same-ontologies	0,11	0,12	0,12

Table 4. The results of AOT and OATL on Multifram track of OAEI 2014.

3 General comments

This is the first time that our systems participate in different tracks of the OAEI 2014 evaluation campaign, and AOT and AOTL are new on the SEALS Platform. However we can conclude with this first participation that AOT provides globally good results in terms of F-measure. Contrary to AOT, the AOTL system provides good results in terms of F-measure on benchmark track but in other tracks the results are not so good.

3.1 Comments on the results

The AOT and AOTL systems are an automatic ontology matching system designed in order to find the correspondence between different entities of ontologies to be aligned. The results obtained by running our systems on different tracks of OAEI 2014 evaluation campaign are slightly good on some tracks but not satisfactory in others.

3.2 Discussions on the way to improve the proposed system

The objective behind the implementation of AOT system is to find the best strategy of aggregation and filter as we have proposed in section 1.2.1.3 (a local filter). Contrary to AOT, the objective behind the implementation of AOTL system is to discover new semantic correspondences by adding other matchers. For now, we have used matchers at terminological and linguistic level.

As we have mentioned before AOT and AOTL systems use terminological information and when these ontologies do not contain this information our two systems fails. Our both systems does not deal with ontologies written in different

languages, and we hope in the future add a module to translate them in the same language.

Another point to be discussed is how to make our systems flexible i.e. the choice of thresholds for the various matchers and ontologies. It is obvious that we cannot set the threshold for all ontologies, in order to find automatically the correspondences between entities of ontologies to be aligned; because each ontology possesses its own specific characteristic?

4 Conclusion

This is the first time the AOT and AOTL systems have participated in OAEI campaign. In this year, our systems have participated in different tracks of OAEI 2014 evaluation campaign.

The AOT system combines the various string-based matching algorithms with average aggregation method. Then we have applied a filter on the combined matrix for the selection of semantic correspondences between ontologies to be aligned. The use of these algorithms is justified by the fact that in the ontologies the terminological information is very important.

Contrary to AOT system, AOTL add at linguistic level an external resource dictionary WordNet for better selection of semantic correspondences.

Finally the results show that our systems can provide some good results. We have used a local filter (in section 1.2.1.3) for AOT system and we envision to study the AOT behavior using different aggregation and filter methods in order to propose in the future new other metrics of filter and aggregation. For the AOTL system, we are interested in the discovery of semantic correspondences by matchers rather than the combination of similarities. We envision using other matchers such as structure-based and reasoning-based matchers in AOTL system.

References

1. J. Euzenat and P. Valtchev, “Similarity-based ontology alignment in owllite,” in Proceedings of ECAI, (2004).
2. J. Euzenat and P. Shvaiko. OntologyMatching. Springer (2007).
3. M. Ehrig. Ontology Alignment Bridging the Semantic Gap. Springer (2007).
4. 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).
5. A. Khiat et M. Benaissa: "Nouvelle Approche d’Alignement d’Ontologies à base d’Instances : trasferet des instances par l’inférence", In The Proceeding of International Conference On Artificial Intelligence and Information Technology, ICA2IT 2014, Ouargla, Algeria, (2014).
6. V. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. Soviet Physics Doklady, 10:707-710, (1966).
7. W. Winkler. The state of record linkage and current research problems. Statistics of Income Division, Internal Revenue Service, Publication R99/04, (1999).

8. M. Ehrig and Y. Sure, “Ontology mapping - an integrated approach,” in Proceedings of the European Semantic Web Symposium ESWS, (2004).
9. G. A. Miller, R. Beckwith, C. D. Fellbaum, D. Gross, K. Miller. WordNet: An online lexical database. Int. J. Lexicograph. 3, 4, pp. 235–244, (1990).

InsMT / InsMTL Results for OAEI 2014 Instance Matching

Abderrahmane Khiat[1], Moussa Benaissa[1]

[1] LITIO Lab, University of Oran, BP 1524 El-Mnaouar Oran, Algeria
abderrahmane_khiat@yahoo.com
moussabenaissa@yahoo.fr

Abstract. InsMT and InsMTL are automatic instance-based ontology alignment systems which (a) annotate instances as first step. In the second step, the InsMT system (b) applies different terminological matchers with a local filter on these annotated instances. Contrary to InsMT, the InsMTL system (b) matches the annotated instances not only at terminological level but also at linguistic level. For the first version of our systems and the first participation at OAEI 2014 evaluation campaign, the results are good in terms of recall but they are not in terms of F-measure.

1 Presentation of the system

1.1 State, purpose, general statement

The instance matching aims to identify similar instances among different ontologies. The systems InsMT (**Ins**tance **M**atching at **T**erminological level) and InsMTL (**Ins**tance **M**atching at **T**erminological and **L**inguistic level) are realized for this purpose. InsMT and InsMTL are automatic instance-based ontology alignment that generates as output an alignment which that contains all the semantic correspondences found between the instances of different concepts of the two ontologies to be aligned.

The InsMT and InsMTL systems annotate the instances as first step with concept and property names.

As second step InsMT uses various string-based matching algorithms i.e. terminological level, these similarities calculated by each algorithm are represented in matrix. InsMT applied a local filter on each matrix, and combines these new similarities with average aggregation method.

Contrary to InsMT, InsMTL system calculates similarities between annotated instances not only at terminological level but also at linguistic level. InsMTL combines the similarities calculated by the various string-based matching algorithms at terminological level, with similarities calculated using an external resource WordNet i.e. at linguistic level. The next step consists in combining the similarities

by gives the priority to linguistic matcher otherwise we have used an average aggregation method.

Finally both systems applied a filter in order to select the semantic correspondences between instances of different ontologies.

The details of each step of InsMT and InsMTL systems are described in the following section.

1.2 Specific techniques used

The process of InsMT and InsMTL systems consists in the following two successive steps: 1) Annotation and Calculation of Similarities and 2) Combination and Extraction of Alignment.

A. InsMT system

1.2.1 Step 1: Annotation and Calculation of Similarities

1.2.1.1 Phase 1: Extraction of Entities of the Ontologies

In this phase, our system takes as input the two ontologies to be aligned and extract their instances.

1.2.1.2 Phase 2: Annotation of Instances

In this phase, our system annotates in this second step the instances with the name and label of the concept also with property name. The purpose of this annotation is to enrich the instances with terminological information. This step is very import especially when instances do contain terminological information.

1.2.1.3 Phase 3: The Applied Matchers

In this phase, our system calculates the similarities between instances, annotated 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.

1.2.2 Step 2: Combination and Extraction of Alignment

1.2.2.1 Phase 1: Local Filter

In this first phase of the second 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.

1.2.2.2 Phase 2: Aggregation of Similarities

In this phase, 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.

1.2.2.3 Phase 3: Global Filter and Identification of Alignment

In this final phase, 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.

B. InsMTL system

We mention in this section the difference between InsMT and InsMTL system.

First, we have added another matcher at linguistic level for InsMTL system in second phase "The applied Matchers", we have used an external dictionary WordNet.

In second step, InsMT does not apply a local filter (phase 1.2.2.1), the similarities calculated by each matcher are represented in matrix without a local filter.

In the phase "Aggregation of Similarities", InsMTL system gives priority to WordNet i.e. if the similarity value calculated using WordNet is greater than the similarity value calculated using string matching algorithms, the similarity value of the matrix combined is equal to the similarity calculated using WordNet, else we use the average aggregation method. The result of the aggregation is represented in a matrix.

1.3 Adaptations made for the evaluation

We do not have made any specific adaptation for the first version of InsMT and InsMTL, for OAEI 2014 evaluation campaign.

1.4 Link to the system and parameters file

The first version of InsMT and InsMTL systems submitted to OAEI 2014 can be downloaded from seal-project at http://www.seals-project.eu/.

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

The results of InsMT and InsMTL systems can be downloaded from seal-project at http://www.seals-project.eu/.

2 Results

In this section, we present the results obtained by running InsMT and InsMTL on instance matching track of OAEI 2014 evaluation campaign.

2.1 Instance Matching

The instance matching track aims at evaluating tools able to identify similar instances among different RDF and OWL ontologies. Our both systems annotate the instances with concept and property names as a first step. Then as second step, the InsMT system uses various string-based matching algorithms on annotated instances in order to find correspondences between them and the InsMTL system use another matcher at linguistic level in order to select semantic correspondences between instances of different concepts.

The table 1 and table 2 below present the results obtained by running InsMT and InsMTL on the instance matching track of OAEI campaign 2014.

2.2.1 Identity Recognition Task

The goal of the id-rec task is to determine when two OWL instances describe the same real-world entity.

Identity Recognition Task	Precision	Recall	F-measure
InsMT	0.0008	0.7785	0.0015
InsMTL	0.0008	0.7785	0.0015

Table 1. The results of InsMT and InsMTL on the Identity Matching track of OAEI 2014.

2.2.2 Similarity Recognition Task

The goal of the sim-rec task is to evaluate the degree of similarity between two OWL instances, even when the two instances describe different real-world entities.

Identity Recognition Task	F-measure
InsMT	d(InsMT) = 37.03

Table 2. The results of InsMT on the Similarity Matching track of OAEI 2014.

3 General comments

3.1 Comments on the results

This is the first time that our systems participate in instance matching track of the OAEI 2014 evaluation campaign, and our InsMT and InsMTL systems are new on the SEALS Platform. However they provide good result in terms of recall but not good result in terms of F-measure.

3.2 Discussions on the way to improve the proposed system

The InsMT and InsMT are automatic instance-based ontology matching systems designed in order to find the correspondence between instances of different concepts.

The objective behind the implementation of InsMT and InsmTL systems is first to find the best strategy of annotation. The InsMT system applied different strategy of aggregation and filter as we have proposed in section in section 1.2.1.3 (a local filter). Contrary to InsMT, the objective behind the implementation of AOTL system is to discover more new semantic correspondences by adding other matchers. For now, we have used matchers at terminological and linguistic level.

As we have mentioned before InsMT and InsMTL systems use terminological information for annotation and matching, and when these ontologies do not contain this information our two systems fails. Our both systems does not deal with instances of ontologies written in different languages, and we hope in the future add a module to translate them in the same language.

Another point to be discussed is how to make our systems flexible i.e. the choice of thresholds for the various matchers (terminological and linguistic). It is obvious that we cannot set the threshold for all instances, in order to find automatically the correspondences between instances of ontologies to be aligned; because each ontology contain instances and possesses its own specific characteristic.

4 Conclusion

This is the first time that InsMT and InsMTL have participated at SEAL platform and OAEI 2014. The InsMT and InsMTL are instance-based ontology alignment system, and in this year, our both systems have participated in instance matching track of OAEI 2014 evaluation campaign.

Initially AOT and AOTL systems annotate instances with concept and property names. The purpose of this annotation is to enrich the instances with terminological information.

The InsMT system calculates similarities between these annotated instances using various string-based matching algorithms. The similarities (between these annotated

instances) calculated by these different matchers are combined using average aggregation after we have applied a local filter on each matrix.

The InsMTL calculates similarities between these annotated instances using the terminological and linguistic matchers. The similarities (between these annotated instances) calculated by these different matchers are combined using average aggregation with the priority to linguistic matcher.

As final step both systems applied a filter on the combined matrix for the selection of semantic correspondences between different instances of different concepts of ontologies.

Finally the results show that our systems provide good results in terms of recall but they are not in terms of F-measure. We envision to select the best aggregation and filtering strategy and add other matchers such as structure-based and reasoning-based matchers.

References

1. J. Euzenat and P. Valtchev, “Similarity-based ontology alignment in owllite,” in Proceedings of ECAI, (2004).
2. J. Euzenat and P. Shvaiko. OntologyMatching. Springer (2007).
3. M. Ehrig. Ontology Alignment Bridging the Semantic Gap. Springer (2007).
4. 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).
5. A. Khiat et M. Benaissa: "Nouvelle Approche d’Alignement d’Ontologies à base d’Instances : trasferet des instances par l’inférence", In The Proceeding of International Conference On Artificial Intelligence and Information Technology, ICA2IT 2014, Ouargla, Algeria, (2014).
6. V. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. Soviet Physics Doklady, 10:707-710, (1966).
7. M. Ehrig and Y. Sure, “Ontology mapping - an integrated approach,” in Proceedings of the European Semantic Web Symposium ESWS, (2004).
8. G. A. Miller, R. Beckwith, C. D. Fellbaum, D. Gross and K. Miller. WordNet: An online lexical database. Int. J. Lexicograph. 3, 4, pp. 235–244, (1990).
9. M. A. Rodriguez and M. J. Egenhofer: Determining Semantic Similarity among Entity Classes from Different Ontologies. IEEE Transactions on Knowledge and Data Engineering, vol. 15, issue 2, pp, 442—456, (2003).
10. 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).
11. A. Maedche, B. Motik, N. Silva and R. Volz "Mafra—a mappingframework for distributed ontologies", Springer, Benjamins VR (eds) EKAW, Berlin, vol 2473, pp 235–250, (2002).
12. K. Todorov, P. Geibel, KU. Kuhnberger "Mining concept similarities for heterogeneous ontologies", Springer, Berlin, ICDM, vol 6171. , pp 86–100, (2010).
13. 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).
14. E. Rahm “Towards large-scale schema and ontology Alignment”, *ReCALL*, (2011).
15. J. Li, J. Tang, Y. Li and Q. Luo, “Rimom: a dynamic multistrategy ontology alignment framework”, *IEEE Trans Knowl*, (2009).

LogMap family results for OAEI 2014 *

E. Jiménez-Ruiz[1], B. Cuenca Grau[1], W. Xia[2], A. Solimando[3], X. Chen[2], V. Cross[2], Y. Gong[1], S. Zhang[1], and A. Chennai-Thiagarajan[2]

[1] Department of Computer Science, University of Oxford, Oxford UK
[2] Computer Science and Software Engineering, Miami University, Oxford, OH, United States
[3] Dipartimento di Informatica, Università di Genova, Italy

Abstract. We present the results obtained in the OAEI 2014 campaign by our ontology matching system LogMap and its variants: LogMap-C, LogMap-Bio 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 fifth participation in the OAEI and the experience has so far been very positive.

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 [13] 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 [17].

LogMap [12, 14] 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 sophisticated 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 [12, 14] 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].

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

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. [6]).

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 [7]. Such encoding, although incomplete, allows LogMap to detect unsatisfiable classes soundly and efficiently.

Axiom tracking and greedy repair. 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.

Semantic indexation. The Horn propositional representation of the ontology modules and the mappings are 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 [5, 19]. In particular, this semantic index allows us to answer many entailment queries over the input ontologies and the mappings computed thus far as an index lookup operation, 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 Adaptations made for the 2014 evaluation

In the OAEI 2014 campaign we have participated with 3 additional variants:

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

LogMap-C is a variant of LogMap which, in addition to the consistency and locality principles, also implements the conservativity principle (see details in [21, 20]). The repair algorithm is more aggressive than in LogMap, thus we expect highly precise mappings but with a significant decrease in recall.

LogMap-Bio includes an extension to use BioPortal [10, 11] as a (dynamic) provider of mediating ontologies instead of relying on a few preselected ontologies [4]. In the OAEI 2014, LogMap-Bio uses the top-5 mediating ontologies given by the algorithm presented in [4]. Note that, LogMap-Bio only participates in the biomedical tracks. In the other tracks the results are expected to be the same as LogMap.

LogMap's algorithm described in [12, 14] has also been adapted with the following new functionalities:

i **Multilingual support.** We have implemented a multilingual module based on *google translate*[4] to participate in the Multifarm track. Additionally, in order to split Chi-

[4] Currently we use the (unofficial) API available at `https://code.google.com/p/google-api-translate-java/`.

nese words, we rely on the ICTCLAS library[5] developed by the Institute of Computing Technology of the Chinese Academy of Sciences.

ii **Extended repair algorithm.** We have extended the Horn propositional projection of the input ontologies to involve data and object properties in the repair process [24]. LogMap's repair module is now more complete and it is also able to repair (object and data) property mappings.[6]

iii **Extended interactive support.** The interactive algorithm described in [14] has been slightly extended to include object and data properties in the process. Note that this extension was already included in the OAEI 2013 campaign.

1.2 Link to the system and parameters file

LogMap is open-source and released under GNU Lesser General Public License 3.0.[7] Latest components and source code are available from the LogMap's Google code page: `http://code.google.com/p/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, with more than 1,500 requests processed so far coming from a broad range of users.

1.3 Modular support for mapping repair

Only 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 [15, 9].

2 Results

In this section, we present a summary of the results obtained by the LogMap family in the OAEI 2014 campaign. Please refer to `http://oaei.ontologymatching.org/2014/results/index.html` for complete results.

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

[6] The OAEI 2014 coherence results does not exhibit these improvements since only the conference track ontologies involve mappings among properties and LogMap 2013 was already coherent. It does have, however, an impact when repairing other mapping sets as shown in [24].

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

Table 1: Results for Benchmark track.

System	biblio			cose			dog		
	P	F	R	P	F	R	P	F	R
LogMap	0.40	0.40	0.40	0.38	0.41	0.45	0.96	0.15	0.08
LogMap-C	0.42	0.41	0.40	0.39	0.41	0.43	0.98	0.15	0.08
LogMapLt	0.43	0.46	0.50	0.37	0.43	0.50	0.86	0.71	0.61

Table 2: Results for Anatomy track.

System	P	F	R	Time (s)
LogMap-Bio	0.888	0.897	0.906	535
LogMap	0.918	0.881	0.846	12
LogMap-C	0.975	0.802	0.682	22
LogMapLt	0.962	0.829	0.728	5

2.1 Benchmark track

Ontologies in this track have been synthetically generated. The goal of this track is to evaluate the matching systems in scenarios where the input ontologies lack important information (e.g., classes contain no meaningful URIs or labels) [8].

Table 1 summarises the average results obtained by LogMap and its variants. Note that the computation of candidate mappings in LogMap (and its variants) heavily relies on the similarities between the vocabularies of the input ontologies; hence, there is a direct negative impact in the cases where the labels are replaced by random strings. Surprisingly, LogMapLt obtained the best results in the dog test case.

2.2 Anatomy track

This track involves the matching of the Adult Mouse Anatomy ontology (2,744 classes) and a fragment of the NCI ontology describing human anatomy (3,304 classes). The reference alignment has been manually curated [25], and it contains a significant number of non-trivial mappings.

Table 2 summarises the results obtained by the LogMap family. LogMap-Bio ranked 2nd in the track. The use of BioPortal as mediating ontology provider had a significant improvement in recall. LogMap-Bio runtime is near 10 minutes since the discovery of the mediating ontologies is performed on-the-fly [4]. Regarding mapping coherence, only two tools (apart from LogMap, LogMap-C and LogMap-Bio) generated coherent alignments. The evaluation was run on a server with 3.46 GHz (6 cores) and 8GB RAM.

2.3 Conference track

The Conference track uses a collection of 16 ontologies from the domain of academic conferences [23]. These ontologies have been created manually by different people and

Table 3: Results for Conference track.

System	RA1 reference			RA2 reference		
	P	F	R	P	F	R
LogMap	0.80	0.68	0.59	0.76	0.63	0.54
LogMap-C	0.82	0.67	0.57	0.78	0.62	0.52
LogMapLt	0.73	0.59	0.50	0.68	0.54	0.45

Table 4: Results for Multifarm track.

System	Different ontologies			Same ontologies		
	P	F	R	P	F	R
LogMap	0.80	0.40	0.28	0.94	0.41	0.27

are of very small size (between 14 and 140 entities). The track uses two reference alignments RA1 and RA2. RA1 contains manually curated mappings between 21 ontology pairs, while RA2 also contains composed mappings based on the alignments in RA1.

Table 3 summarises the average results obtained by the LogMap family. The last column represents the total runtime on generating all 21 alignments. Tests were run on a laptop with Intel Core i5 2.67GHz and 8GB RAM. LogMap ranked 2nd and LogMap-C ranked 3rd. They both produced coherent alignments.

2.4 Multifarm track

This track is based on the translation of the OntoFarm collection of ontologies into 9 different languages [18].

In the OAEI 2014, only LogMap, AML and XMap implemented specific multilingual techniques. Table 4 summarises the results. LogMap achieved very competitive results in terms of precision. Regarding recall, however, there is still room for improvement. In the close future we plan to extend the multilingual module with more sophisticated translation techniques.

2.5 Library track

The library track involves the matching of the STW thesaurus (6,575 classes) and the TheSoz thesaurus (8,376 classes). Both of these thesauri provide vocabulary for economic and social sciences. Table 5 summarises the results obtained by the LogMap family. The track was run on a computer with one 2.4GHz core with 7GB RAM and 2 cores. LogMap ranked 2nd in this track. The results for LogMap* are obtained with a version of the input OWL ontologies using skos labels (i.e. *skos:altLabel* and *skos:prefLabel*).

2.6 Interactive matching track

The interactive track is based on the conference track and it uses the RA1 reference alignment as Oracle. Table 6 summarizes the obtained results by LogMap with the

Table 5: Results for Library track.

System	P	R	F	Time (s)
LogMap*	0.743	0.711	0.681	223
LogMap	0.775	0.705	0.648	74
LogMapLt	0.644	0.703	0.771	9
LogMap-C	0.484	0.342	0.264	22

Table 6: Results for Interactive track.

System	RA1 reference			Avg. Calls	Time (s)
	P	R	F		
LogMap	0.88	0.73	0.64	4	27

Table 7: Summary results for the Large BioMed track

System	Total Time (s)	P	F	R	Inc. Degree.
LogMap	1,751	0.890	0.792	0.719	0.013%
LogMap-Bio	8,634	0.843	0.784	0.744	0.8%
LogMap-C	6,331	0.907	0.688	0.559	0.013%
LogMapLt	317	0.868	0.613	0.532	34.0%

interactive mode activated. LogMap with interactivity improved both the average Precision and Recall wrt LogMap with the interactive mode deactivated (see Section 2.3). LogMap performed on average, 3.91 calls to the Oracle along the 21 matching tasks. LogMap ranked 2nd in the interactive matching track, but it was the system performing less calls to the oracle.

2.7 Large BioMed track

This track consists of finding alignments between the Foundational Model of Anatomy (FMA), SNOMED CT, and the National Cancer Institute Thesaurus (NCI). These ontologies are semantically rich and contain tens of thousands of classes. UMLS Metathesaurus [3] has been selected as the basis for the track reference alignments.

Table 7 summarises the results obtained by the LogMap family. The table shows the total time in seconds to complete all tasks in the track and averages for Precision, Recall, F-measure and Incoherence degree. The track was run on a Ubuntu Laptop with an Intel Core i7-4600U CPU @ 2.10GHz x 4 and allocating 15Gb of RAM..

Only AML and LogMap variants (excluding LogMapLt) generated almost coherent alignments. LogMap ranked 2nd in the track, while LogMap-C and LogMap-Bio obtained the best average Precision and the second best average Recall, respectively. LogMapLt was the fastest to complete all tasks.

Table 8: Results for OA4QA track.

System	Queries	RA1 reference			RAR1 reference		
		P	F	R	P	F	R
LogMap	18/18	0.750	0.741	0.750	0.729	0.728	0.750
LogMapC	18/18	0.722	0.704	0.694	0.722	0.703	0.694
LogMapLt	11/18	0.409	0.379	0.423	0.351	0.348	0.402

Table 9: Results for Instance matching track.

System	Identity		
	P	F	R
LogMap	0.603	0.099	0.054
LogMap-C	0.642	0.078	0.042

2.8 OA4QA track

The Ontology Alignment for Query Answering (OA4QA) track [22] does not follow the classical ontology alignment evaluation with respect to a set of reference alignments. Precision and recall is 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. Given a query and an ontology pair, a model (or reference) answer set is computed using the correspondent reference alignment for the ontology pair. Precision and recall is calculated with respect to these model answer sets.

In the OAEI 2014 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. Table 8 summarises the (average) results for the LogMap family. LogMap and LogMap-C ranked 1st and 2nd in the track, although the number of queries is still not large enough to provide representative values for Precision and Recall. However, the most interesting result is the number of queries a system is able to answer when the computed alignments is incoherent. For example, LogMapLt, since it does not implement mapping repair techniques, is only able to answer 11 of the queries, which damages the obtained precision and recall.

2.9 Instance matching track

The results of LogMap (and LogMap-C) were not as good as previous years. Note that, LogMap does not implement specialised instance matching techniques. Nevertheless, LogMap outperformed two of the participating tools specialised in instance matching. Table 9 summarises the results obtained by LogMap and LogMap-C.

3 General comments and conclusions

3.1 Comments on the results

LogMap, apart from Benchmark and Instance Matching tracks for which does not implement specific techniques, has been one of the top systems in the OAEI 2014. Fur-

thermore, it has also been one of the few systems implementing repair techniques and providing (almost) coherent mappings in all tracks.

LogMap's main weakness relies on the fact that the computation of candidate mappings is based on the similarities between the vocabularies of the input ontologies; hence, there is a direct negative impact in the cases where the ontologies are lexically disparate or do not provide enough lexical information (e.g. Benchmark and Instance Matching).

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. There are, however, many exciting possibilities for future work. For example we aim at improving the 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[8] [16], which presents a very challenging scenario.

3.3 Comments on the OAEI test cases

The number and quality of the OAEI tracks is growing year by year. However, there is always room for improvement:

Comments on the OA4QA track. The new OA4QA track has succesfully shown the negative impact of a incoherent alignment in query answering tasks. However, the number of queries is still small to provide representative values for the F-measure. More queries and more challenging ontologies will make the track more attractive.

Comments on the OAEI interactive matching track. The interactive track has been a very important step forward in the OAEI, however, larger and more challengings tasks should be included. For example, matching tasks (e.g. anatomy and largebio) where the number of questions to the expert user or Oracle may be critical. Furthermore, it is quite unlikely that the expert user will be perfect, thus, the interactive matching track should also consider the evaluation of several Oracles with different error rates such as the evaluation performed in [14].

Comments on the OAEI largebio track. One of the objectives of the largebio track is the creation of a "silver standard" reference alignment by harmonising the output of the different participating systems. In the next OAEI campaign it would be very interesting to actively use this "silver standard" in the construction of the track's reference alignment. This will help to improve the completeness of the reference alignment.

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)

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

2. Baeza-Yates, R.A., Ribeiro-Neto, B.A.: Modern Information Retrieval. ACM Press / Addison-Wesley (1999)
3. Bodenreider, O.: The unified medical language system (UMLS): integrating biomedical terminology. Nucleic Acids Research 32, 267–270 (2004)
4. 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)
5. 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)
6. Cuenca Grau, B., Horrocks, I., Kazakov, Y., Sattler, U.: Modular reuse of ontologies: Theory and practice. J. Artif. Intell. Res. 31, 273–318 (2008)
7. 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)
8. Euzenat, J., Rosoiu, M.E., dos Santos, C.T.: Ontology matching benchmarks: Generation, stability, and discriminability. J. Web Sem. 21, 30–48 (2013)
9. 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)
10. 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)
11. 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)
12. Jiménez-Ruiz, E., Cuenca Grau, B.: LogMap: Logic-based and Scalable Ontology Matching. In: Int'l Sem. Web Conf. (ISWC). pp. 273–288 (2011)
13. 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)
14. 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)
15. Jiménez-Ruiz, E., Meilicke, C., Cuenca Grau, B., Horrocks, I.: Evaluating mapping repair systems with large biomedical ontologies. In: 26th Description Logics Workshop (2013)
16. 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)
17. Meilicke, C.: Alignment Incoherence in Ontology Matching. Ph.D. thesis, University of Mannheim (2011)
18. Meilicke, C., Castro, R.G., Freitas, F., van Hage, W.R., Montiel-Ponsoda, E., de Azevedo, R.R., Stuckenschmidt, H., Šváb-Zamazal, O., Svátek, V., Tamilin, A., Trojahn, C., Wang, S.: MultiFarm: a benchmark for multilingual ontology matching. J. Web Sem. (2012)
19. Nebot, V., Berlanga, R.: Efficient retrieval of ontology fragments using an interval labeling scheme. Inf. Sci. 179(24), 4151–4173 (2009)
20. 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)
21. 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)
22. Solimando, A., Jiménez-Ruiz, E., Pinkel, C.: Evaluating Ontology Alignment Systems in Query Answering Tasks. In: Poster at Int'l Sem. Web Conf. (ISWC) (2014)
23. Šváb, O., Svátek, V., Berka, P., Rak, D., Tomášek, P.: OntoFarm: towards an experimental collection of parallel ontologies. In: Int'l Sem. Web Conf. (ISWC). Poster Session (2005)
24. Zhang, S., Jiménez-Ruiz, E., Cuenca Grau, B.: Inconsistency Repair in Ontology Matching. MSc thesis., University of Oxford (2014), http://www.cs.ox.ac.uk/isg/projects/LogMap/papers/Master_thesis_Shuo_Zhang.pdf
25. Zhang, S., Mork, P., Bodenreider, O.: Lessons learned from aligning two representations of anatomy. In: Conf. on Princliples of Knowledge Representation and Reasoning (KR) (2004)

Alignment Evaluation of MaasMatch for the OAEI 2014 Campaign

Frederik C. Schadd, Nico Roos

Maastricht University, The Netherlands
{frederik.schadd, roos}@maastrichtuniversity.nl

Abstract. This paper summarizes the results of the fourth participation of the MaasMatch system in the Ontology Alignment Evaluation Initiative (OAEI) competition. We describe the performed changes to the MaasMatch system and evaluate the effect of these changes on the different datasets.

1 Presentation of the system

MaasMatch is a ontology mapping system with the initial focus of fully utilizing the information located in the concept names, labels and descriptions in order to produce a mapping between two ontologies [2,4]. This was achieved through the utilization of syntactic similarities and virtual documents, which can also be used as a disambiguation method for the improvement of lexical similarities [3,6].

1.1 Specific techniques used

The 2014 version of *MaasMatch* exhibits some notable changes compared to the 2013 version [5]. First, the system is now based on a de-centralized *configuration* system. For each presented mapping problem, the system queries its stored similarity measures whether the current problem is appropriate for that particular measure. Each measure independently evaluates whether the given ontologies contain a sufficient amount of exploitable input data and whether the ontologies have an appropriate size. The measures then report their results back to the system. As an example, the instance similarity would not consider itself appropriate if one of the given ontologies does not contain any instances. Additionally, each similarity also evaluates the size of the input ontologies, such that computationally expensive similarities are not executed on large-scale problems.

Using all similarities that have responded positively for the current problem, the system computes the similarity cube between the two ontologies. Here, all similarity measures are executed in *parallel* using a dynamic number of threads depending on the current hardware, such that the system can scale with the number of available computing cores. This facilitates the computation of alignments between large-scale ontologies through a more effective usage of all available computing power.

The resulting similarity cube is aggregated using the Dempster-Shafer theory, after which the result alignment is extracted. The entire mapping process of the *MaasMatch* system is visualized in Figure 1.

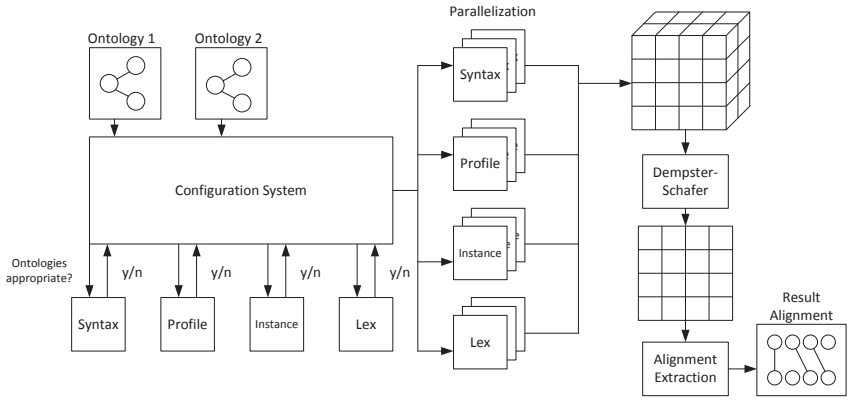


Fig. 1. Visualization of the MaasMatch architecture.

1.2 Adaptations made for the evaluation

While the system can provide correspondences with a wide range of confidence values, we have applied a hard threshold to the result extraction such that the evaluation on track which do not perform thresholding analysis better reflect the actual quality of the alignments. However, the applied threshold can easily be adjusted in the supplied configuration file.

1.3 Link to the system and parameters file

MaasMatch and its corresponding parameter file is available on the SEALS platform and can be downloaded at http://www.seals-project.eu/tool-services/browse-tools.

2 Results

This section presents the evaluation of the OAEI2014 results achieved by MaasMatch. When applicable, the performance of this year will be compared to the performance of the previous year [1]

2.1 Benchmark

The benchmark track consists of synthetic datasets, where an ontology is procedurally altered in various ways and to different extents, in order to see under what circumstances a system can still produce good results. Table 1 displays the results on the two evaluated datasets:

Compared to the results of the previous year [5], the performance of *MaasMatch* saw a shift towards the precision of the alignments. While the precisions of the previous year were in the range of 0.6, this year the precisions of the different benchmark

Test Set	Precision	F-Measure	Recall
biblio	0.97	0.56	0.37
cose	0.98	0.48	0.31
dog	0.92	0.55	0.39

Table 1. Harmonic means of the benchmark test sets.

ontologies ranged from 0.92 to 0.98. This came however at a slight cost of recall. The likely reason behind this is the re-introduction of a hard-threshold which is applied after the alignment extraction step.

Another interesting point of note is that, compared to the previous year [1], the testing procedure no longer caused issues in the execution of the system.

2.2 Anatomy

The anatomy dataset consists of a single matching task, which aligns a biomedical ontology describing the anatomy of a human to an ontology describing the anatomy of a mouse. Unique aspects about this ontology are their large sizes and the fact that they contains specialized vocabulary which is not often found in non-domain specific thesauri. Table 2 displays the results of this dataset.

Year	Precision	F-Measure	Recall	Runtime(s)
2013	0.359	0.409	0.476	8532
2014	0.914	0.803	0.716	49

Table 2. Results of the anatomy data set.

On the anatomy test track we can observe some significant improvements compared to last year's evaluation. First, we can see a significant improvement with regard to the alignment quality. Both the precision and recall have improved drastically compared to the previous year, with an absolute increase of 0.555 and 0.24 respectively. Additionally, the runtime for this dataset has been reduced drastically. Both the configuration system, which would not execute complex similarities (e.g. the lexical similarity), and the parallelized execution of all similarities contribute to this increase.

2.3 Conference

The confidence data set consists of numerous real-world ontologies describing the domain of organizing scientific conferences. The results of this track can be seen in Table 3.

Year	Precision	F-Measure	Recall
2013(ra1)	0.28	0.37	0.55
2014(ra1)	0.64	0.55	0.48
2013(ra2)	0.27	0.36	0.53
2014(ra2)	0.52	0.50	0.49

Table 3. Results of the conference data set.

Overall we can observe an improved performance on the conference dataset for both the *ra1* and *ra2* reference alignments. The likely reason behind this is the improved selection and aggregation of the similarity measures. The runtime of the entire evaluation was 68777 seconds. This is significantly higher than the runtime of the anatomy track, since the conference track consists of numerous small mapping tasks. The system analyses each task individually with regard to its complexity. For the anatomy track, the single task is evaluated as too large for time consuming similarity measures, such that these are dropped. However, any given mapping problem of the conference track is small enough such that the application of time consuming similarities is still feasible, resulting in the overall runtime being higher for this track than for the larger anatomy track.

2.4 Multifarm

The Multifarm data set is based on ontologies from the OntoFarm data set, that have been translated into a set of different languages in order to test the multi lingual capabilities of a specific system. The results of MaasMatch on this track can bee seen in Table 4.

Year	Precision	F-Measure	Recall
2013 (same ontology)	0.62	0.29	0.19
2014 (same ontology)	0.52	0.10	0.06
2013 (different ontology)	0.01	0.02	0.03
2014 (different ontology)	0.27	0.15	0.10

Table 4. Results of the multi-farm data set.

Despite the system not being designed for multi-lingual mapping, we saw an improvement in performance for the mapping tasks with different ontologies. For this part of the dataset, the precision was increased significantly while the recall saw a moderate increase.

For mapping tasks consisting of the same ontology being translated into different languages the overall performance was lower than the previous year. A likely reason for

this is that the internal structures of the concepts are no longer taken into consideration compared to last year, such that a decreased performance for mapping problems with identical structures are to be expected.

2.5 Large BioMed

The Large Biomedical track consists of three mapping problems in which very large ontologies modelling the biomedical domain have to be mapped. The results of this track can be seen in table 5.

Year	Precision	F-Measure	Recall	Runtime(s)
2013 (FMA-NCI Task 1)	0.407	0.456	0.517	12,409
2014 (FMA-NCI Task 1)	0.808	0.824	0.840	1,460
2013 (FMA-SNOMED Task 1)	-	-	-	-
2014 (FMA-SNOMED Task 1)	0.655	0.664	0.674	4,605

Table 5. Results of the multi-farm data set.

We can observe some significant improvements compared to the results of the previous year. In the previous year, *MaasMatch* was unable to produce a result alignment within the set time limit for the FMA-SNOMED matching task. This year, the system did produce an alignment within the time limit with a F-measure of 0.664. The results for the FMA-NCI track have improve significantly. Both the precision and recall have improved over the previous year, resulting in an increase of F-Measure from 0.456 to 0.824. In addition, the required runtime for this task has been reduced by approximately 89%.

Some issues however remain for this dataset. Further improvements need to be made such that the system can tackle the largest task (NCI-SNOMED). This year, the system was unable to complete this task due to memory issues. The likely cause behind this is the current implementation of the profile similarity. To improve runtime, this similarity caches all concept profiles in memory such that these do not have to be re-created whenever a similarity computation is invoked on the same concept. However, due to the large size of the matching tasks this optimization is no longer a feasible solution due to memory constraints.

3 General comments

3.1 Comments on the results

Overall, we have seen improved results for all evaluation tracks, leading to competitive performances when compared to the other mapping systems. Furthermore, large-scale mapping problem were now solvable within a reasonable runtime for the first time. Some weaknesses still remain, for instance the result alignments being non-coherent, but ultimately the current iteration of the system has been largely successful.

3.2 Discussions on the way to improve the proposed system

The *MaasMatch* system saw some significant changes compared to last year's iteration, which is reflected in the different results for all tracks. Most observed changes were indeed positive. Some areas however remain were more improvements can be made. This year saw the introduction of a decentralized self-configuration system, where the logic of determining whether a similarity measure is appropriate is de-coupled to each particular metric. There current implementation however is only preliminary. We foresee an improved system which contains a set of testing problems, similar to the different tracks of OAEI, on which every similarity metric can be automatically evaluated with regard to its compatibility and run-time efficiency. These results could then be stored and consulted for any new mapping task.

Currently, multi-lingual problems are not supported. While we did investigate the possibility of multi-lingual adaptations, none of the available options were satisfactory. On-line solutions, e.g. *Google Translate* have the issue that these are typically commercial, such that there are no free options for research available, and limited with regard to the amount of queries on can issue per month, making the adoption for large-scale problems infeasible. *Off-line* options, such as *BabelNet* have the issue that these are much larger than the available storage per system on the SEALS platform (5.1GB as opposed to the 500MB limit). A solution would be to establish a private server on which *BabelNet* can be queried by the system, though this was not pursued due to time constraints.

4 Conclusion

In this paper we presented the results of the *MaasMatch* system for the 2014 OAEI campaign. The system has changed significantly compared to the previous year, which is reflected in the performance of the different tracks. Overall, most tracks have seen improvements with regard to alignment quality. The self-configuration system now made the mapping of large problem in a feasible time a possibility, as evidence in the runtime performance during the anatomy track.

References

1. Bernardo Cuenca Grau, Zlatan Dragisic, Kai Eckert, Jérôme Euzenat, Alfio Ferrara, Roger Granada, Valentina Ivanova, Ernesto Jiménez-Ruiz, Andreas Oskar Kempf, Patrick Lambrix, et al. Results of the ontology alignment evaluation initiative 2013. In *Proc. 8th ISWC workshop on ontology matching (OM)*, pages 61–100, 2013.
2. Frederik C. Schadd and Nico Roos. Maasmatch results for oaei 2011. In *Proceedings of The Sixth International Workshop on Ontology Matching (OM-2011) collocated with the 10th International Semantic Web Conference (ISWC-2011)*, pages 171–178, 2011.
3. Frederik C. Schadd and Nico Roos. Coupling of wordnet entries for ontology mapping using virtual documents. In *Proceedings of The Seventh International Workshop on Ontology Matching (OM-2012) collocated with the 11th International Semantic Web Conference (ISWC-2012)*, pages 25–36, 2012.

4. Frederik C. Schadd and Nico Roos. Maasmatch results for oaei 2012. In *Proceedings of The Seventh ISWC International Workshop on Ontology Matching*, pages 160–167, 2012.
5. Frederik C. Schadd and Nico Roos. Summary of the maasmatch participation in the oaei-2013 campaign. In *Proceedings of The Eighth International Workshop on Ontology Matching (OM-2013) collocated with the 12th International Semantic Web Conference (ISWC-2013)*, pages 139–145, 2013.
6. Frederik C Schadd and Nico Roos. Word-sense disambiguation for ontology mapping: Concept disambiguation using virtual documents and information retrieval techniques. *Journal on Data Semantics*, pages 1–20, 2014.

OMReasoner: Combination of Multi-matchers for Ontology Matching: results for OAEI 2014

Guohua Shen, Yinling Liu, Fei Wang, Jia Si, Zi Wang, Zhiqiu Huang, Dazhou Kang

College of Computer Sci. &Tech., Nanjing Univ. of Aeronautics and Astronautics, Nanjing, China
{ghshen,ylliu,fwang,jsi,zwang,zqhuang,dzkang}@nuaa.edu.cn

Abstract. Ontology matching produces correspondences between entities of two ontologies. The **OMReasoner** is unique in that it creates an extensible framework for combination of multiple individual matchers, and reasons about ontology matching by using external dictionary *WordNet* and description logic reasoner. It handles ontology matching in both literal and semantic level, and it makes use of the semantic part of OWL-DL as well as structure. This paper describes the result of **OMReasoner** in the OAEI 2014 competition in three tracks: benchmark, conference, and MultiFarm.

1 Presentation of the system

Ontology matching finds correspondences between semantically related entities of the ontologies. It plays a significant role in many application domains.

Many approaches to ontology matching have been proposed: the implementation of match may use multiple match algorithms or matchers, and the following largely-orthogonal classification criteria are considered [1-3]: schema-level and instance-level, element-level and structure-level, syntactic and semantic, language-based and constraint-based.

Many approaches focus on syntactic aspects instead of semantic ones. OMReasoner achieves the matching by means of some external dictionary and reasoning techniques. Still, this approach includes strategy of combination of (mainly syntactical) multi-matchers (e.g., EditDistance matcher) before match reasoning.

1.1 State, purpose, general statement

The matching process can be defined as a function *f*.

$$A' = f(O1, O2, A, p, r)$$

Where *O1* and *O2* are a pair of ontologies as input to match, *A* is the input alignment between these ontologies and *A'* is new alignment returned, *p* is a set of parameters (e.g., weight *w* and threshold τ) and *r* is a set of oracles and resources.

Alignments express correspondences between two entities. A correspondence must express two corresponding entities and the relation that is supposed to hold between them. Given two ontologies, a correspondence is a 5-tuple:<id,e1,e2,R,n>, where

. id is a unique identifier of the given correspondence;

. e1 and e2 are the entities of the first and the second ontology respectively;

. R is a relation (e.g., equivalence(=), more general(>), less general(<), disjointness ⊥)) holding between e1 and e2. In OAEI campaign, equivalence is mainly considered;

. n is a confidence measure (typically in the [0 1] range) for the correspondence between e1 and e2.

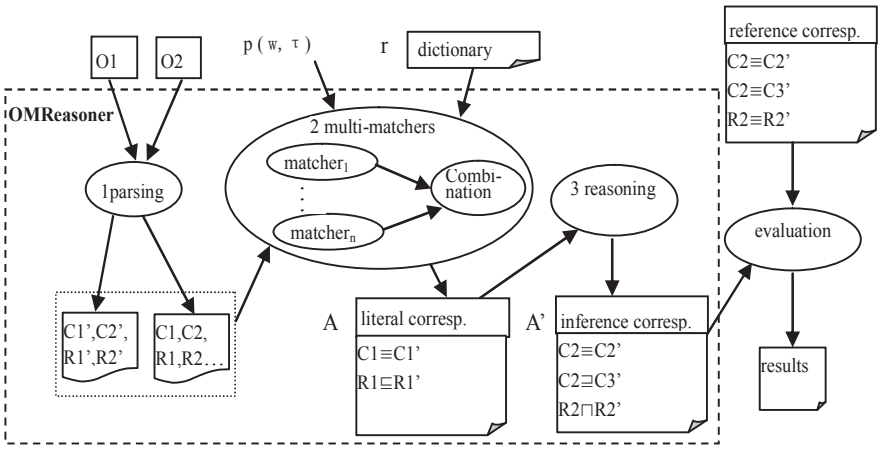


Fig.1. Ontology matching in OMReasoner

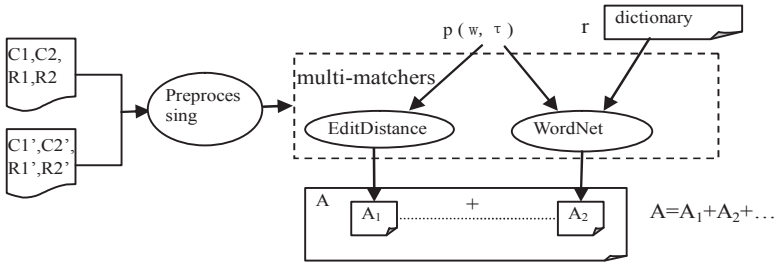


Fig.2. Instances of multi-matchers in OMReasoner

The OMReasoner achieved ontology alignment as following three steps (see Fig.1):

1. Parsing: we can achieve the classes and properties of ontologies by using ontology API: Jena.
2. Combination of multiple individual matchers: the literal correspondences (e.g. equivalence) can be produced by using multiple match algorithms or matchers, for example, string similarity measure (prefix, suffix, edit distance) by string-based, constrained-based techniques. Meanwhile, some semantic correspondences can be achieved by using an external dictionary: WordNet. Then the multiple match results can be combined by using specific strategy.

The framework of multi-matchers combination is supported, which facilitates inclusion of new individual matchers.

3. Reasoning: the further semantic correspondences can be deduced by using DL reasoner, which uses literal correspondences produced in step 2 as input.

Finally, we evaluate the results against the reference alignments, and compute two measures: precision and recall.

In OMReasoner, the framework for multi-matchers is flexible, and any new individual matcher can be included. Now, the instances of multi-matchers include *EditDistance* and *WordNet* (see Fig.2).

1.2 Specific techniques used

1. Threshold

Threshold is necessary for many matchers (especially syntactic ones) to determine whether the similarity is regarded as equivalence. For example, the edit distance of "book" and "booklet" is 3/7 (i.e., the similarity confidence measure is 1-3/7=0.57). If the threshold is 0.55, then these two entities are equivalent (with confidence measure 0.57); else if threshold is 0.6, they are not. So, we have to tune our tool via threshold.

2. Combination of confidence measure

Each individual matcher can produce correspondences with confidence measures. All these confidence measures will be normalized before combination. OMReasoner includes following flexible strategies to combine the multiple match results:

(a) weighted summarizing algorithm (WeightSum)

The confidence can be summarized by weighted similarity algorithm (see formula 1), where w_k is the weight for a specific matcher k, and $sim_k(e1,e2)$ is the confidence measure of similarity (mainly equivalence) by this method.

$$sim(e1, e2) = \sum_{k=1}^{n} w_k \times sim_k(e1, e2), \quad \text{where } \sum_{k=1}^{n} w_k = 1.0 \tag{1}$$

(b) maximum method (Max)

The maximum confidence measure is chosen among n matchers (see formula 2) .

$$sim(e1, e2) = \max(sim_1(e1, e2), \ldots, sim_n(e1, e2)) \tag{2}$$

3. semantic matching

OMReasoner uses semantic matching methods like *WordNet* matcher and description logic (DL) reasoning.

WordNet[1] is an electronic lexical database for English, where various senses (possible meanings of a word or expression) of words are put together into sets of synonyms. Relations between ontology entities can be computed in terms of bindings between WordNet senses. This individual matcher uses an external dictionary: WordNet to achieve semantic correspondences.

[1] http://wordnet.princeton.edu/

OMReasoner employs DL reasoner provided by Jena. OMReasoner includes external rules to reason about the ontology matching. However, reasoning is time consuming and only contributes a little to results. In this version, reasoning is skipped.

2 Results：a comment for each dataset performed

In this section, we present the results obtained by OMReasoner in the OAEI 2014 competition. It participated in three tracks: benchmark, conference, and MultiFarm. Tests were carried out on a PC running Windows Server 2008 R2 Standard with Intel Core i5 processor running at 2.8 Ghz and 16 GB RAM.

2.1 Benchmark

In this track, the ontologies can be divided into 3 categories(see Table 1) . In group 1, the lexical information have been altered to change their labels or identifiers. This alteration includes replacing the labels or identifiers with other names that follow a particular naming convention, a random name, a misspelled name or a foreign word. In group 2 have ontologies that have flattened hierarchies, expanded hierarchies or no hierarchies at all. In group 3 the ontologies are the most challenging ones to align. This is because labels have been scrambled such that they comprise a permutation of letters of a particular length. We tune our tool by using threshold T and combination strategy S, then get the better results (τ_{wd}=0.95, τ_{ed}=0.9; S=Max). The results obtained by OMReasoner in the benchmarks track are summarized in Table 2.

Table 1. The categories of the Benchmark 2014

category	concept	systematic			real ontology
tests cases	101-104	201-210	221-247	248-266	301-304

Table 2. Results for the Benchmark 2014

	101-104	201-210	210-247	248-266	301-304	H-mean
precision	0.898	0.675	0.820	0.637	0.925	0.791
recall	1.000	0.414	1.000	0.517	0.437	0.647
F-measure	0.946	0.491	0.898	0.555	0.574	0.694

2.2 Conference

The confidence data set consists of numerous real-world ontologies describing the domain of organizing scientific conferences. We use Combination strategy to run our system tool in Conference track. The results obtained by OMReasoner in the benchmarks track are summarized in Table 3 (τ_{wd}=0.9, τ_{ed}=0.8**; S=Max**).

Table 3. Results for the Conference 2014

test case	precision	recall	F-measure
Conference	0.778	0.518	0.647

2.3 MultiFarm

MultiFarm track is composed of a subset of the Conference dataset, translated in eight different languages. In this track, the ontologies can be divided into 2 categories. In group 1 the alignments ontologies are the same. In group 2 the alignments ontologies are different.

Firstly, we take use of dictionary to translate different languages into English. Then, the translated English is imported in multi-matchers by using Max strategy. Finally we get the results. We tune our tool by using threshold, and the results can be seen in Table 4(τ_{wd}=0.8, τ_{ed}=0.6; S=Max), which show that the F-Measures of the ontologies alignments in group 2 are obviously worse than those in group 1. We think the reasons are that OMReasoner is not well designed to match different ontologies which are written in completely different languages yet.

Table 4. Results for the MultiFarm 2014

Test case	precision	recall	F-measure
Group 1: Same ontologies	0.955	0.800	0.853
Group 2: Different ontologies	0.584	0.438	0.471

To choose better threshold, we compare the results (see Table 5) across several thresholds in Conference track. Still we use Max method to run our tool. From the results, we find that when threshold τ_{wd}=0.9, τ_{ed}=0.8, our tool performs best. So that we take use of threshold τ_{wd}=0.9, τ_{ed}=0.8 in Conference track. Using the same method, we get the better thresholds for Benchmark and MultiFarm track.

Table 5. Comparison results of different thresholds for the Conference 2014

Threshold		precision	recall	F-measure
τ_{wd}	τ_{ed}			
0.8	0.8	0.782	0.508	0.599
0.95	0.8	0.787	0.466	0.580
0.9	0.8	0.778	0.518	0.647
0.9	0.9	0.796	0.476	0.580

3 General comments

3.1 Discussions on the way to improve the proposed system

The performance of inference relies on the literal correspondences heavily, so more accurate results which are exported from multi-matchers will greatly enhance the results of our tool. Some approaches to improving our tool are listed as follows:

1. Adopt more flexible strategies in multi-matchers combination instead of just weighed sum.
2. Add some preprocessing (see Fig.2), such as eliminating specific character (e.g., '-', '_') or separating compound words, before words are imported into matchers.
3. Take comments and label information of ontology into account, especially when the name of concept is meaningless.
4. Reexamine the use of an appropriate threshold value to optimize accuracy.

Another problem in our tool is that we ignore structure information among ontology at the present stage. And we will improve it in the future.

3.2 Comments on the OAEI procedure

OAEI procedure arranged everything in good order, furthermore SEALS platform provides a uniform and convenient way to standardize and evaluate our tool.

3.3 Comments on the OAEI test cases

The OAEI test cases involve all kinds of fields which include conference, anatomy, language, etc. The variety of tracks and the improvements introduced along the years makes the campaign very useful to test the performance of ontology aligners and analyses their strengths and weaknesses. Nevertheless, we miss blind tests cases in more tracks, which will allow a fair comparison between systems.

3.4 Proposed new measures

After serious discussion, we believe that OMReasoner can improve a lot. Some new ways proposed as follows:

1. Enrich the semantic dictionaries because WordNet is not a professional dictionary, which cannot obtain more comprehensive semantic concepts.
2. Take into account hierarchical ones instead of only all concepts and properties.
3. Find NCI thesaurus for anatomy track.
4. Find different languages dictionaries for MultiFarm.
5. Improve the algorithm of some matchers.
6. Include more different matchers.

4 Conclusions

In this paper, we presented the results of the OMReasoner system for aligning ontologies in the OAEI 2014 competition in three tracks: benchmark, conference, and MultiFarm. The combination strategy of multiple individual matchers and DL reasoner are included in our approach. This is the third time we participate the OAEI, the results are still not satisfying and we will improve it in the future.

References

1. Rahm, E. and Bernstein, P.: A survey of approaches to automatic schema matching. *The VLDB Journal*, ,10(4): 334--350(2001).
2. Shvaiko, P. and Euzenat, J.: A survey of schema-based matching approaches. *Journal on Data Semantics (JoDS)* IV, 146--171(2005).
3. Kalfoglou, Y. and Schorlemmer, M.: Ontology mapping: the state of the art. *The Knowledge Engineering Review Journal*, 18(1):1--31, (2003).
4. Shvaiko, P.: Iterative Schema-based Semantic Matching. PhD, University of Trento, (2006)
5. Jian, N., Hu, W., Cheng, G. et al: Falcon-AO: Aligning Ontologies with Falcon. *In: Proceedings of the K-CAP Workshop on Integrating Ontologies* (2005)
6. Do, H. and Rahm, E.: COMA- a system for flexible combination of schema matching approaches. *In: Proceedings of the International Conference on Very Large Databases*, 610--621.(2002)
7. Giunchiglia, F., Shvaiko, P., and Yatskevich, M.: S-Match: an algorithm and an implementation of semantic matching. *In: Proceedings of the European Semantic Web Symposium*, 61--75.(2004)
8. Kalfoglou, Y. and Schorlemmert, M.: If-map: an ontology mapping method based on information flow theory. *In: Proceedings of ISWC'03, Workshop on Semantic Integration*, (2003)
9. Bouquet, P., Serafini, L., and Zanobini, S.: Semantic coordination: A new approach and an application. *In: Proceedings of the International Semantic Web Conference*, 130--145.(2003)
10. Baader, F., Calvanese, D., McGuinness, D., et al.: The description logic handbook: theory, implementations and applications. *Cambridge University Press*, (2003)
11. Ehrig, M., Sure, Y.: Ontology mapping - an integrated approach. *In Proceedings of the European Semantic Web Symposium (ESWS)*, 76--91, (2004)
12. RacerPro User Guide. http://www.racer -systems. com, 2005
13. Do, H., Melnik, S., Rahm, E.: Comparison of Schema Matching Evaluations. *In: Proceedings of the 2nd Intl. Workshop on Web Databases*, Erfurt, Germany:,221--237(2002)
14. Shen, G., Jin, L., Zhao, Z., Jia, Z., He, W., and Huang, Z. : OMReasoner: using reasoner for ontology matching: results for OAEI 2011. *In Proceedings of* the 6th International Workshop on Ontology Matching.
15. Shen, G., Tian, C., Ge, Q., Zhu, Y., Liao, L., Huang, Z., and Kang D.: OMReasoner: using multi-matchers and reasoner for ontology matching: results for OAEI 2012. *In Proceedings of* the 7th International Workshop on Ontology Matching.

RiMOM-IM Results for OAEI 2014

Chao Shao, Linmei Hu, Juanzi Li

Tsinghua University, Beijing, China.
{shaochao,hulinmei,ljz} @ keg.tsinghua.edu.cn

Abstract. This paper presents the results of RiMOM-IM in the Ontology Alignment Evaluation Initiative (OAEI) 2014. We only participated in IM@OAEI2014. We first describe the overall framework of our matching System (RiMOM-IM); 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-IM.

1 Presentation of the system

Recently, a number of ontological knowledge bases have been built and published, such as DBpedia[1]. , YAGO [2], Xlore [3], etc. Some published knowledge bases are domain specific ones that cover facts within one domain, such as movie, music and geography; some other ones are cross-domain knowledge bases that contain various kinds of information in different domains. Usually, knowledge about one object may be contained in different knowledge bases. For example, both YAGO and elvisPedia contain information about a person named "Elvis Presley"; YAGO records the birthdate of this person while elvisPedia has the information about his wife; if we want know more about "Elvis Presley", we have to search his information in different knowledge bases. Therefore, there is a growing need to align different knowledge bases so that we can easily get more complete knowledge about things that we are interested in.

A lot of work has already been done for aligning ontological knowledge bases. Previous researches focus on aligning the schema elements (i.e. concepts and properties) in knowledge bases, which is called ontology matching. Most recently, the problem of matching instances in different knowledge bases has attracted increasing interest. Many instance matching approaches have been proposed. Our system is proposed for large-scale instance matching. There are two major techniques in the existing approaches to speed up the instance matching process: blocking and iterative matching. Blocking is to index the instances in two knowledge bases separately and then select the instances having the same keys as candidate instance pairs. Iterative matching is to find the instance correspondences in multiple loops; only a fraction of instances are matched in each iteration, which are then used as seeds for matching the rest instances in the following iterations. Although the above two techniques are very helpful to large-scale instance matching, there are still several challenging problems which are not well addressed. First, since usually only literal values in RDF triples are used as indexing keys for blocking, the set of candidate instance pairs to be compared is still very large. Second, iterative instance matching is likely to propagate minor errors of mismatched instances in each iteration. Traditional decision-making methods can hardly get rid of

mismatched instances since instances in two different knowledge bases are usually described by different numbers of RDF triples.

In order to solve the above challenges in large-scale instance matching, we propose an iterative instance matching framework RiMOM-IM (RiMOM-Instance Matching), which is developed based on our ontology matching system RiMOM [4]. The main idea behind the framework is to maximize the utilization of distinctive and available matching information. RiMOM-IM presents a novel blocking method to improve the efficiency and employs a weighted exponential function based similarity aggregation method to guarantee high accuracy of instance matching.

1.1 State, purpose, general statement

This section describes the overall framework of RiMOM-IM. The overview of the instance matching system is shown in Fig. 1. The system includes five modules, i.e., *Initial Interactive Configuration*, *Candidate Pair Generation*, *Matching Score Calculation*, *Instance Alignment* and *Validation*. The annotated numbers in the figure show the sequences of the process. We illustrate the process as follows.

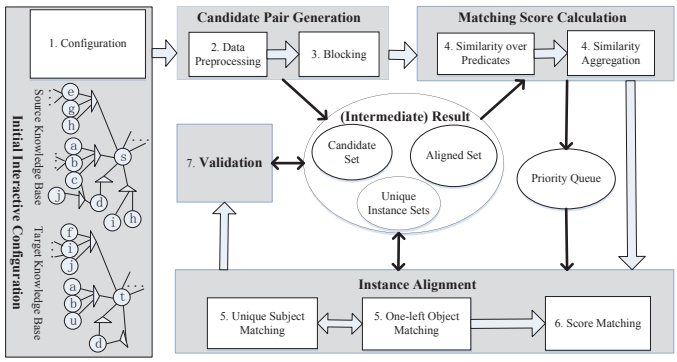


Fig. 1. Framework of RiMOM-IM

1. The system begins with *Initial Interactive Configuration*, which allows users to configure system with needed modules in the following process and their parameters.
2. We conduct data preprocessing, such as unifying data formats for the values of some predicates.
3. We proceed blocking which consists in using inverted indexing to generate *candidate set* and *unique instance sets*.
4. For each pair in the *candidate set*, we compute similarities over all aligned predicates with ”Similarities over Predicates” and then through ”Aggregation”, we aggregate them to get the final matching score of two instances. We generate a priority queue by sorting the final scores in a descending order.

5. For *unique instance sets*, we iteratively use "Unique Subject Matching" and "One-left Object Matching" to generate *aligned set* until no new aligned instances are generated. These aligned instances will then be used to find new candidate pairs and new unique instances, thus updating *candidate set* and *unique instance sets*. Correspondingly, matching scores for related instance pairs and the priority queue will be updated.
6. For the priority queue, we use "Score Matching" to generate only one aligned pair with the highest score above the threshold. If there is a newly aligned instance pair, we will generate new unique instances, which will be taken as input to step 5. If there is no new aligned pair, we continue step 7.
7. If Validation module is chosen in step 1, we will conduct validation on all aligned pairs. Otherwise, terminate.

1.2 Specific techniques used

This year we only participate in the IM@2014 track. We will describe specific techniques used in this track.

Data Preprocessing: First, we translate all the languages used in the whole datasets to English by using google translator. Then we remove special symbols like "♯, *, !", etc. and stop words like "a, of, the", etc. Afterwards, we calculate the TF-IDF values of words in each knowledge base.

Blocking: Blocking 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 blocking method, we take the predicate as well as top 10 words of the object (ordering by tf-idf values in the knowledge base) as index keys of instances. It should be noticed that if the object is an instance, the entire URI is considered as a word. Owing to the novel blocking method which restricts the candidate pairs with identical distinctive information (predicate and distinctive object features), we greatly reduce the number of similarity comparisons and improve the efficiency.

Similarity over Predicates: The similarity function varies with different predicates. For example, we can use indicator function for the predicate of birthdate, when the value are the same, the indicator is 1, otherwise, 0. For the predicate of comments, we compute cosine similarity based on the tf-idf vectors. In system configuration, we can specify a similarity function for each aligned predicate.

Similarity Aggregation: For each instance pair, after acquiring similarity values in terms of multiple aligned predicates, we need to aggregate the similarities to get final matching score. AVG aggregates the similarities by computing the average value [5]. SIGMOID(SIG) aggregates the similarities by computing the average similarities transformed by a sigmoid function [5]. These methods do not adapt to the case when different instance pairs have different numbers of aligned predicates. In this work, we propose a weighted exponential aggregation function, *ExpAgg* to aggregate the similarities *S*, which is a set of similarities of all aligned predicates. The function is as follows:

$$ExpAgg(S) = \frac{\Sigma_{s_i \in S} w_i' * exp(w_i'' * s_i)}{\Sigma_{s_i \in S} w_i' * exp(w_i'' * 1)} \quad (1)$$

Among them, s_i is the similarity score in terms of the i^{th} aligned predicate. We set the weights of the predict "label" and the other predicts as as 16 and 1, respectively.

Score Matching: In this task, we don't use the modules of "Unique Subject Matching" and "One-left Object Matching". Each time we choose the pair with the highest score as the aligned pair, we will then update the matching score of each instance pair in the prior queue. With the greedy algorithm of extracting only the most matching pair every time, we control error propagation to some extent. As we can not guarantee a global optimization with the greedy algorithm, we add the later process of validation.

Validation:Since many objects of instances are URIs referring to other instances, there still exists some nondeterminacy in aligning two instances due to the uncertainty in the alignment situation of their $compatible\_neighbors$, and we also find some rules are very useful. We add validation module to correct some mistakes by some useful rules. In this track, we find that if two instances both contain the predict of "lable", their "label" predicts shall share at least a same token.

1.3 Link to the system and parameters file

The RiMOM-IM system can be found at `http://keg.cs.tsinghua.edu.cn/project/RiMOM/`.

2 Results

The IM@2014 track contains two subtasks. we present the results and related analysis for the two subtasks in the following subsections.

2.1 Identity Recognition sub-task

The goal of the Identity Recognition sub-task is to determine whether two OWL instances refer to the same real-object. Due to a lack of training data, it is very difficult for us to tuning our parameters. First, we use the default setup to get a preliminary result, and then we check the information of some aligned pairs. We find out that the predict of "label" is very important, so we increase the weight of the "label" predict. Finally, we get 1103 instance pairs as matching ones.

As show in figure 2, the results for the identity task are: Precision 0.65, Recall 0.49, Fmeasure 0.56, which is much lower than we expected. But we are pretty sure that if we have some training set, we can tuning a much better result.

2.2 Similarity Recognition sub-task

The goal of the Similarity Recognition sub-task is to determine the degree of similarity between two OWL instances, even when the two instances describe different real-objects. In our system, we use the traditional cosine similarity measurement, however, if one predicate have many similarity values, we use the maximum value. So in summary, we use maxpooling+cosine similarity. We can find that if two instances describe the same real-objects, their similarity value will usually be larger than that of two instances

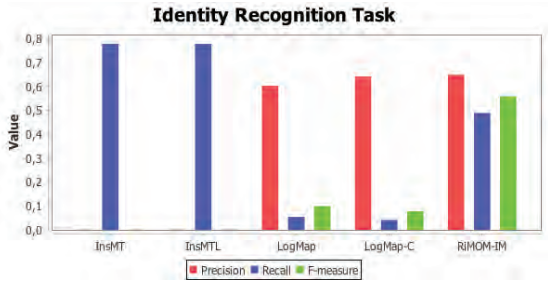


Fig. 2. Results of the Identity Recognition Task

which describe different real-objects. Therefore, it's reliable to use the similarity value as a measure to judge whether two instances describe the same real-objects. And we find that if two instances describe different real-objects while have a high similarity value, then their labels are usually different. We can use these two observations for instance matching. As show in figure 3, this similarity strategy is much close to the crowdsourc-

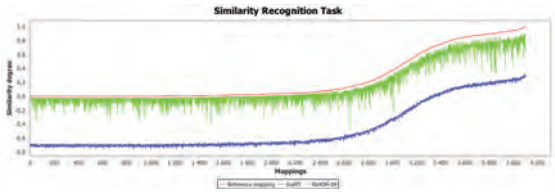


Fig. 3. The Result of Similarity Recognition Task

ing activities'. We have chosen other complexity similarity measurements, but it turns out that this simple measurements works better.

2.3 Discussions on the way to improve the proposed system

Our system need the aligned predicates to select the candidate instance pairs. Our system will use the aligned instance pairs to calculate the similarity values of other instance pairs, which will also need the information of aligned predicates. We need to invent an algorithm to automatically align the predicates. Although there are some algorithms that can align the instances by measuring the similarity values of predicates, none of them use the aligned instance pairs to help to update the similarity values for predicates. We will develop an algorithm that do not need any aligned predicates, but can iteratively use the aligned instance pairs to align the predicates, which will in turn advance the instance alignment.

2.4 Comments on the OAEI 2014 measures

This task is a cross-lingual instance matching task. We find out that we can significantly improve the result by using translation method. And we find that the blocking method also improves the precision of the result. Because the "datatype" of the "object" is always "String", we do not have any relations between any two instances. So we can't use the relation information to improve the recall. This year we use ten keywords for every predict to get more candidate pairs to ensure a high recall.

3 Conclusion and future work

In this paper, we present the system of RiMOM-IM in OAEI 2014 Campaign. We participate in one track this year. We described specific techniques we used during this campaign. In our project, we design a new framework to do the instance matching task. Our method effective and efficient.

For now, we need to tune the parameter manually, we will improve it by making the tuning process automatic in the future work.

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. 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
5. Jean-Mary, Y.R., Shironoshita, E.P., Kabuka, M.R.: Ontology matching with semantic verification. J. Web Sem. **7**(3) (2009) 235–251

RSDL Workbench Results for OAEI 2014*

Simon Schwichtenberg[1], Christian Gerth[2], and Gregor Engels[1]

[1] University of Paderborn, s-lab – Software Quality Lab, Germany
{simon.schwichtenberg,engels}@upb.de
[2] Osnabrück University of Applied Sciences, Germany
c.gerth@hs-osnabrueck.de

Abstract The RSDL workbench was developed as a part of a service composition platform for service markets and provides tools to specify structural and behavioral aspects of services based upon the Rich Service Description Language (RSDL). Such comprehensive service descriptions allow a multi-faceted matching of service requests and offers in terms of their data models, operations, and protocols. Domains and application contexts of such service requests and offers are not known to the matchers in advance. Our data model matcher exploits several background ontologies to find corresponding data model elements. Data model alignments are represented in the form of relational Query View Transformation (QVT) scripts that are used to normalize behavioral models, which is a prerequisite for operation matching. For the OAEI campaign, we excluded background ontologies, because the involved additional costs did not justify the gain yet. In this paper, we present our system and the results for the OAEI campaign.

1 Presentation of the system

RSDL Workbench (RSDLWB) is a collection of tools for the specification, discovery and composition of services. A service discovery brings service requesters and providers together by matching requirements and existing services. These requirements or the offered functionality can be described in terms of the structural as well as behavioral aspects of the service through RSDL [4]. An RSDL specification consists of a data model, operation signatures, Visual Contracts (VCs) [2], and protocols. For the specification of a service, a data model determines relevant data types and their relationships in terms of a Unified Modeling Language (UML) class model. A VC is typed over such and specifies the behavior of certain operations. In particular, a VC describes pre- and postconditions of operation calls by graph grammar rules whose graphs conform to the class model.

The domain(s) of the service requests and offers are not known to the matcher in advance. Even though they share the same domain and describe semantically equivalent concepts, their respective class models might be heterogeneous, because they are created independently most likely. VCs might be heterogeneous as well, due to the heterogeneity of their respective class models they conform to. Consequently, VCs of a requester and a provider and hence the behavior of service offers and requests cannot be compared directly and must be normalized.

* This work was partially supported by the German Research Foundation (DFG) within the Collaborative Research Centre "On-The-Fly Computing" (SFB 901)

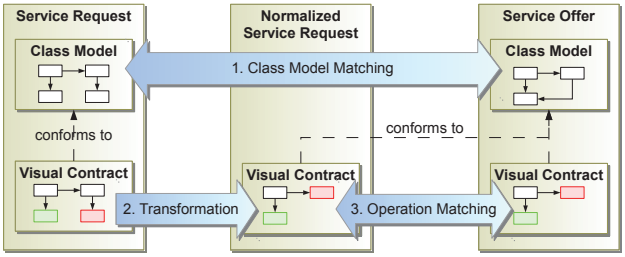


Figure 1: Matching Process [9]

Fig. 1 gives an overview of our approach: (1) The class models are matched and a list of class, attribute, and association mappings is returned. (2) Based on the list of mappings, a relational QVT [1] model transformation script is automatically generated which allows bidirectional model transformations. The VCs of the requester are normalized according to the providers' class model by executing the model transformation. The normalization of the VCs is a prerequisite for the operation matching. (3) Once all VCs conform to the same class model, they can be compared directly. In a next step, the operations are matched based on the normalized VCs, which is explained in detail in [5].

The list of operation mappings is the input for the protocol matcher, that checks if the operation invocation sequences requested by the requester match with the operation invocation sequences allowed by the provider. The data model matcher and the transformation script generation was previously presented in [9]. The system was realized as an Eclipse plug-in and implements the interface of EMF Compare[3] in order to reuse its graphical user interface. This paper focuses on its data model matching techniques and the results of the OAEI campaign.

1.1 State, purpose, general statement

As explained in Sect. 1, the purpose of the system is to match heterogeneous class models. The system automatically matches two UML class models that are part of respective RSDL specifications and generates a relational QVT model transformation script, which acts as a mediator enabling the translation of behavioral models. If necessary, the generated script can be manually revised.

In context of our system, the relevant OAEI tracks that we aim to compete in, are as follows: *benchmark*, *anatomy*, and *conference*. In the future, we also plan to participate in the *multifarm*, *library*, and *largebio* track. The tracks *interactive*, *instance matching*, and *ontology alignment for query answering* are less relevant for RSDLWB and support for these tracks is not scheduled.

In our knowledge, none of the existing matching system fulfills all the requirements of RSDLWB class model matcher, i.e. (1) process UML class models as input, (2) create 1:1, 1:n, n:1, n:m class mappings, (3) generate a transformation script from the mappings.

[3] `http://www.eclipse.org/emf/compare/`

1.2 Specific techniques used

According to the classification of [3], RSDLWB uses the following matching techniques: 1. String-based (normalization), 2. Language-based (tokenization), 3. Constraint-based (type similarity), 4. Linguistic resources / domain specific ontologies (background ontologies)[4], 5. Taxonomy-based (upward cotopic similarity)[4].

RSDLWB matches classes (`DataProperties`), attributes (`DataProperties`), and associations (`ObjectProperties`) pairwise and independently. The similarity of a pair is basically determined on the basis of their labels. In case of attributes, their type similarities [10] are considered as tie breakers. Before labels of two concepts are matched, they are split into *tokens*. Each single token is *normalized* by lowercasing and suppression of non-alphabetical characters. Next, the tokens are matched for their part. The overall label similarity arises from the average similarity of the token matching. If two tokens have identical normalized strings, they are assumed to match and get the highest similarity value.

The rest of this section addresses techniques that were not used in the OAEI campaign for reasons that are explained in Sect. 3. When two tokens are not identical, their Upward Cotopic (UC) similarity [6] is computed. The UC similarity is the quotient of the number of the tokens' shared hypernyms and the number of all their hypernyms according to a Background Ontology (BO). Such a BO is selected when it contains two concepts with the same normalized labels as the tokens to be matched. In particular, an individual BO is selected for each label pair. BOs are stored in a relational database. The transitive closure of the hypernyms is precalculated for each BO concept and also stored in the database. We imported different ontologies to our database like WordNet [8], DBpedia [7], etc.

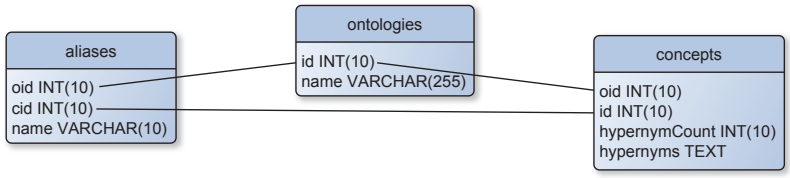


Figure 2: Database Tables and Foreign Key Relations

Fig. 2 shows the database schema: The *ontologies* table contains a row for each imported BO. The *alias* table contains all synonyms for the *concepts*, which are stored in a separate table. The column *hypernyms* stores all hypernyms as a list of concept ids. Additionally, *hypernymCount* contains the number of hypernyms. The edges illustrate foreign keys of the tables. A database index was added to *aliases.name*, which allows faster lookups of hypernyms for inquired aliases.

Listing 1 shows an exemplary SQL query that illustrates how two tokens from the input ontologies are anchored in a BO and how their hypernyms are retrieved. For each pair of tokens, an individual BO is selected. It might happen that a token is anchored in a BO by a homonym. The selection strategy prioritizes BOs with deeper taxonomic

[4] Technique was not used in the OAEI campaign (c.f. Sect. 3)

```
SELECT c1.hypernyms AS hypernyms1, c2.hypernyms AS hypernyms2,
    a1.oid AS id, LEAST(c1.hypernymCount, c2.hypernymCount) AS
    prio FROM aliases AS a1, aliases AS a2, concepts AS c1,
    concepts AS c2 WHERE a1.name = 'person' AND a2.name = '
    author' AND a1.oid=a2.oid AND c1.id=a1.cid AND c2.id=a2.cid
    AND c1.oid=a1.oid AND c2.oid=a2.oid ORDER BY prio DESC LIMIT
     1;
```

Listing 1: Querying Background Ontologies

hypernyms1	hypernyms2	id	prio
160, **843**, **138**, **269**, **515**, **325**, **932**	346, 930, 431, **160**, **843**, **138**, **269**, **515**, **325**, **932**	2	7

Table 1: Query Result

hierarchies, because shallow hierarchies produce UC similarity values that are close to each other. Tab. 1 shows the query result set that contains the hypernyms of *person* and *author*, the ontology id and the priority. Accordingly, the UC similarity is:

$$\sigma_{UC} = \frac{|hypernyms1 \cap hypernyms2|}{|hypernyms1 \cup hypernyms2|} = \frac{7}{10} = 0.7$$

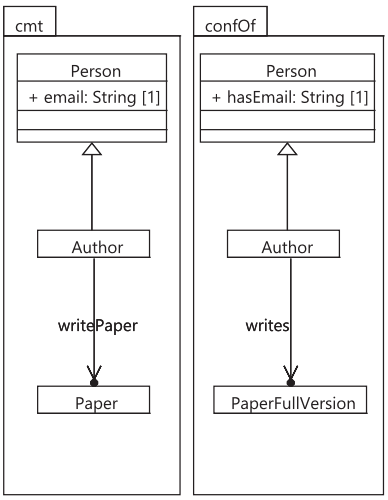
To create n:m class mappings, a simple greedy algorithm is used. At first, the class pairs are sorted in a descending order according to their similarity. The algorithm iterates over the pairs and if none of the current pair's classes is part of a mapping, a new mapping is created. If one class is already part of a mapping and the second is not, the second is added to the mapping the first is already part of. If both classes are part of a mapping, the pair is ignored.

1.3 Generation of the Model Transformation

In this section, we want to explain briefly how the QVT transformation script is generated from the alignment. The generation is exemplified on the basis of the reference alignment for the `cmt` and the `confOf` ontologies that are part of the conference track. The UML diagram in Fig. 3a shows parts of these ontologies and is arranged in a way so that some mappings of the reference alignment can easily be seen.

Fig. 3b shows the generated QVTr script: Each class mapping corresponds to a **top relation**, which is a possible entry point for the transformation, e.g. <*Person, Person*> (line 2). During the transformation, free variables (**domains**) like `person1` are bound to instances of the source class model at first. Accordingly, `var_email` is bound to `person1`'s data attribute `email` (l. 5). The **enforce** keyword directs the transformation to create proper instances in the target data model (if necessary). Once `person2` is bound to a (newly created) instance, its attribute `hasEmail` is bound to `var_email` (l. 8). Variables for object attributes (l. 13, 16) are delegated to other relations to bind free variables (l. 19). The delegation is carried out in **when** clauses, which are preconditions for the **relations**. The creation of the script is not trivial, because n:m mappings have to be considered or mapped attributes do not necessarily belong to classes that have

been mapped for their part, etc. For a more detailed description on the script generation and its current limitations, the reader is referred to [9].



(a) Excerpt of `cmt` and `confOf` Ontologies

```
transformation Cmt_ConfOf(cmt : cmt, confof : confof){
   top relation Person_Person{
      var_email:String;
      enforce domain cmt person1 : Person {
         email = var_email
      };
      enforce domain confof person2 : Person {
         hasEmail = var_email
      };
   }
   top relation Author_Author{
      enforce domain cmt author1 : Author {
         writePaper = var_paper1 : Paper{}
      };
      enforce domain confof author2 : Author {
         writes = var_paper2 : PaperFullVersion{}
      };
      when{
         Paper_PaperFullVersion(var_paper1, var_paper2);
      }
   }
   top relation Paper_PaperFullVersion{
      enforce domain cmt paper1 : Paper{};
      enforce domain confof paper2 : PaperFullVersion{};
   }
}
```

(b) Generated QVTr Script

1.4 Link to the system and provided alignments

The SEALS compliant[5] RSDLWB 1.1 is available at `http://goo.gl/3Uj9gS`. The provided alignments are available at `http://goo.gl/JLsELe`.

2 Results

The RSDLWB results are summarized in Tab. 2. The second column denotes how the values for precision, F-measure, and recall were calculated. The harmonic mean of all test cases is stated for *benchmark*, *conference*, and *multifarm*. The tracks *anatomy* and *library* comprise only one test case. Concerning the *conference* track, the values are calculated according two reference alignments ra1 and ra2. The *multifarm* track has two kind of tasks: The first kind matches the same ontology in different languages (same) and the second different ontologies in different languages (diff). Relating to *largebio*, RSDLWB could only complete the test case FMA-NCI within 10 hours.

2.1 benchmark

The test cases of the *benchmark* track are systematically generated from three seed ontologies – biblio, cose, and dog – by modifying or discarding ontology features. The evaluation is conducted in a blind fashion, i.e. neither the participants nor the organizers

[5] `http://oaei.ontologymatching.org/2014/seals-eval.html`

Track		Runtime [h:m:s]	Precision	F-measure	Recall
benchmark biblio	H-Mean	00:01:26	.99	.66	.5
benchmark dog	H-Mean	04:00:17	.99	.75	.6
anatomy	Mouse-NCI	00:22:17	.978	.749	.607
conference	H-Mean ra1	00:00:36	.81	.59	.47
conference	H-Mean ra2	00:00:36	.76	.54	.42
multifarm	H-Mean (diff)	00:18:00	.16	.04	.02
multifarm	H-Mean (same)	00:18:00	.34	.02	.01
library	TheSoz-STW	09:07:08	.781	.073	.038
largebio	FMA-NCI	00:36:57	.956	.38	.237

Table 2: RSDL Workbench Results for OAEI 2014

know the generated test cases in advance. RSDLWB achieved very good results regarding F-measure for the biblio and dog test cases. However, RSDLWB did not produce an alignment for cose.

2.2 anatomy

The Adult Mouse Anatomy and a part of the National Cancer Institute Thesaurus (NCI) describing the human anatomy are matched in the *anatomy* track. In regard to precision, F-measure, and recall, RSDLWB performs slightly worse than baseline StringEquiv. RSDLWB achieved high precision for the price of low recall compared to other systems.

2.3 conference

In the *conference* track, seven independent ontologies in the domain of organizing conferences are matched pairwise, resulting in 21 test cases. The produced alignments from the participants are evaluated against the reference alignments ra1 and ra2. The reference alignment ra2 is generated as the transitive closure computed on ra1. While ra1 was available to participants, ra2 was not. Regarding F-measure, RSDLWB performs better than baseline StringEquiv, but slightly worse than baseline edna, which means an average performance. Since RSDLWB relies only on string-based techniques the results are similar to the baseline algorithms.

2.4 multifarm

The goal of this track is to evaluate the ability of the matcher to deal with ontologies in different languages. The cross-lingual matching scenario is relevant for RSDLWB, but we did not investigate on this scenario yet. The low precision, F-measure, and recall values result from the fact, that labels in different languages share less common tokens. Even with enabled BOs, the matcher does not support other languages than English at the moment.

2.5 library

The task of the *library* track is to match the STW and the TheSoz thesaurus, which include a huge amount of concepts and additional descriptions. These ontologies define multiple labels per concept in different languages. However, RSDLWB does not support multiple labels per concept yet. Rather, it selects an arbitrary label, so that these labels are possibly in different languages, which leads to the same problems as for multifarm and explains the weak results.

2.6 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. The input size of the ontologies vary across the six test cases. RSDLWB could only complete the smaller FMA-NCI test case within the given time frame of 10 hours. For this particular test case, RSDLWB achieved significantly lower F-measure than the average of all participants.

3 General comments

Several adjustments had been made to enable a participation of the RSDLWB in the OAEI campaign: (1) An abstraction layer for the input models was introduced in order to enable the matching of Web Ontology Language (OWL) ontologies. Since RSDLWB was designed to match UML models, it does not support other OWL features except labels of `Classes`, `DataProperties`, and `ObjectProperties`. (2) The matcher was configured to create only 1:1 mappings instead of n:m mappings, because n:m mappings had a negative impact on the most tracks. (3) Originally, as presented in [9], the matcher partially used some combinatorial algorithms which were replaced by simple greedy algorithms to improve the runtime. (4) The UC similarity was disabled, because the additional lookups of hypernyms in the BOs did not justified the matching results. With enabled UC similarity, more false positives than true positives were created, resulting in a decreased average F-measure.

3.1 Comments on the results

After the first participation of the RSDLWB in the OAEI campaign, we conclude that the system is not optimal for the OAEI test tracks yet and that there were no improvements in any of the OAEI disciplines. As the results show, the matcher heavily relies on labels and rarely on other ontology features. Furhermore, the system in its current shape is not suitable to match large ontologies.

3.2 Discussions on the way to improve the proposed system

RSDLWB depends very much on labels. To overcome this issue, similarity metrics must be introduced that take e.g. structural features of the ontologies into account. Since the importance of the similarity metrics varies between the test tracks and cases, the

matcher should be adaptive and adjust the weights for these metrics. RSDLWB failed to complete test tracks with large ontologies in a reasonable time – even without using BOs. To improve the runtime of the matcher, we plan to parallelize the retrieval of hypernyms and the calculation of similarities. When BOs are used, the system often produces false negatives because it uses homonyms for the anchoring in BOs. Therefore, we want to adjust the matcher so that it is aware of the matching task's domain. Furthermore, we want to address cross-lingual matching by importing multilingual data sets of DBpedia or by integrating a translation service. We are confident that we can improve the system once the BO can be exploited effectively.

4 Conclusion

The first evaluation of RSDL workbench in the OAEI 2014 campaign showed good results for the benchmark track, but average to weaker results for the other tracks. The runtime and the quality of the matching results is improvable compared to other systems. We excluded the usage of background ontologies, because they increase the runtime of the system, but did not improve the matching results on average. As soon as we can effectively exploit BOs, we need to improve the systems' efficiency, because the retrieval of hypernyms has an extra effect on the runtime.

References

1. Meta Object Facility (MOF) 2.0 Query/View/Transformation Specification Version 1.1. `http://www.omg.org/spec/QVT/1.1/PDF/` (January 2011)
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. Euzenat, J., Shvaiko, P.: Ontology Matching, vol. 18. Springer (2007)
4. Huma, Z., Gerth, C., Engels, G., Juwig, O.: A UML-based Rich Service Description Language for Automatic Service Discovery of Heterogeneous Service Partners. In: CAiSE Forum. pp. 90–97 (2012)
5. 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)
6. Maedche, A., Zacharias, V.: Clustering Ontology-based Metadata in the Semantic Web. In: Elomaa, T., Mannila, H., Toivonen, H. (eds.) PKDD 2002, LNCS (LNAI), vol. 2431, pp. 348–360. Springer (2002)
7. Mendes, P.N., Jakob, M., Bizer, C.: DBpedia for NLP: A Multilingual Cross-domain Knowledge Base. In: Proc. of the 8th International Conference on Language Resources and Evaluation (LREC) (2012)
8. Miller, G.A.: WordNet: A Lexical Database for English. Communications of the ACM 38(11), 39–41 (1995)
9. 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)
10. Tibermacine, O., Tibermacine, C., Cherif, F.: WSSim: a Tool for the Measurement of Web Service Interface Similarity. In: Proceedings of the french-speaking Conference on Software Architectures (CAL) (2013)

XMap++ : Results for OAEI 2014

Warith Eddine Djeddi and Mohamed Tarek Khadir

LabGED, Computer Science Department, University Badji Mokhtar, Annaba, Algeria
{djeddi, khadir}@labged.net

Abstract. In this paper, we present the results obtained by our ontology matching system XMap++ within the OAEI 2014 campaign. XMap++ is a scalable ontology alignment tools capable of matching large scale ontology. This is our second participation in the OAEI, and we can see an overall improvement on nearly every task.

1 State, purpose, general statement

XMap (eXtensible Mapping) is an ontology alignment tool for the alignment of OWL entities (i.e., classes, object properties and data properties). XMap++ approach uses different similarity measures of different categories such as string, linguistic, and structural based similarity measures to understand ontologies semantics. A weights vector must, therefore, be assigned to these similarity measures, if a more accurate and meaningful alignment result is favored. Combining multiple measures into a single similarity metric has been solved using weights determined by intelligent strategies [3].

The major drawback of our two previous versions XMapGen and XMapSig [2], despite the fact that they achieved fair results and the aim of their development is to deliver a stable version, the time performance was very low time, especially for the Large Biomedical Ontologies tracks, inability to recognize multiple labels to a single entity as synonyms and inability to recognize labels translated in different languages (e.g Chinese, Czech, Dutch, French, German). After carefully studying this issue, we realize that our algorithm needs more assessment in its performance. This inspires us to consider new strategies in the new version of XMap++ 2014, such as : 1) Using cosine similarity as a string similarity methods to compare the concepts textual descriptions associated with the nodes (labels, names, identity, etc) of each ontology; 2) Involving particular parallel matching on multiple cores or machines for dealing with the scalability issue on ontology matching; 3) Translating labels with different languages using Bing Translator (not use any services which require payment); 4) Interfacing with the Wordnet electronic dictionary using Java Wordnet Interface (JWI) as a Java library. Meanwhile, XMap++ loads WordNet dictionary fully into memory to gain time when it aligns large-scale ontologies. Consequently, the new version XMap++ 2014 has improved both the matching quality and time performance in large scale ontology matching tasks.

1.1 Specific techniques used

The workflow and the main components of the system can be seen in the Fig. 1. The XMap++ consists of the following components:

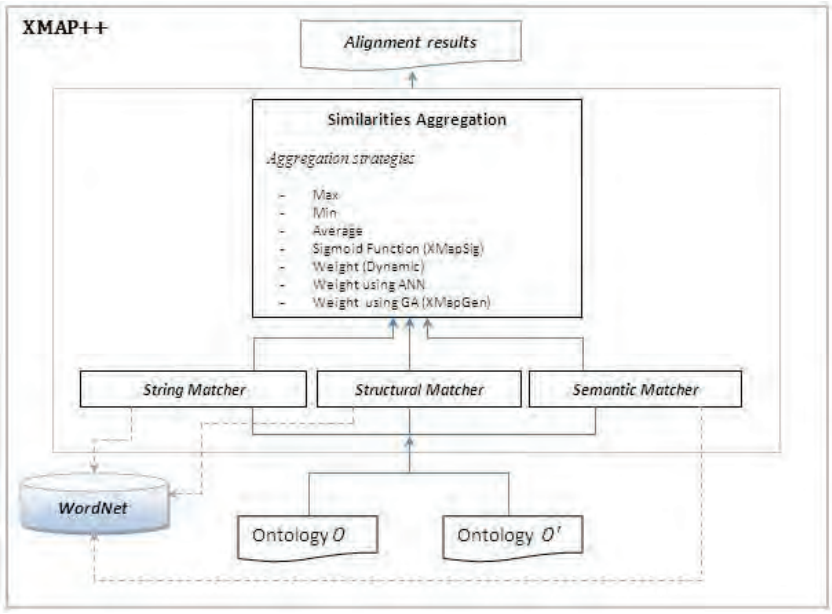


Fig. 1. Sketch of Architecture for XMAP++.

1. Matching inputs are two ontologies, source O and target O' parsed by an Ontology Parser component;
2. The ***String Matcher*** based on linguistic matching compares the textual descriptions of the concepts associated with the nodes (labels, names) of each ontology;
3. The ***Linguistic matcher*** jointly aims at identifying words in the input strings, relaying on WordNet [7]. These matching techniques may provide incorrect match candidates, structural matching is used to correcting such match candidates based on their structural context. In order to deal with lexical ambiguity, we introduce the notion of the *scope* belonging to a concept which represents the context where it is placed [1]. The value of linguistic methods is added to the linguistic matcher or the structure matcher in order to enhance the semantic ambiguity during the comparison process of entity names;
4. The ***structural matcher*** aligns nodes based on their adjacency relationships. The relationships (e.g., *subClassOf* and *is-a*) that are frequently used in the ontology serve, at one hand, as the foundation of the structural matching;
5. The three matchers perform similarity computation in which each entity of the source ontology is compared with all the entities of the target ontology, thus producing three similarity matrices, which contain a value for each pair of entities. After that, an aggregation operator is used to combine multiple similarity matrices computed by different matchers to a single aggregated similarity matrix. We refer to [3] for more detail about the pruning and splitting techniques on data matrices for two couple of entities;

6. XMap++ uses three types of aggregation operator; these strategies are *aggregation*, *selection* and *combination* [3];
7. Finally, these values are filtered using a selection according to a defined threshold and the desired cardinality. In our algorithm, we adopt the *1-1* cardinality to find the optimal solution in polynomial time.

2 Results

In this section, we present the evaluation results obtained by running XMap++ with SEALS client with *Benchmark*, *Anatomy*, *Conference*, *Multifarm*, *Library* 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.

2.1 Benchmark

XMap++ performs very well in terms of Precision (1.0) while a low recall (0.4) in the Benchmark track. Those low values are explained by the fact that ontological entities with scrambled labels, lexical similarity becomes ineffective. Whereas for the others two test suites our algorithm performed worse in term 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	P	R	F
biblio	1.0	0.40	0.57
cose	1.0	0.17	0.28
dog	1.0	0.20	0.32

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. XMap++ achieves a good F-Measure value of ≈89% in an adequate amount of time (22 sec.) (see Table 2). In terms of F-Measure/runtime, XMap++ ranked 3rd among the 10 tools participated in this track.

2.3 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 kinds; this opens a useful way to test our semantic matchers. The match quality was evaluated

Table 2. Results for Anatomy track.

System	Precision	F-Measure	Recall	Time(s)
XMap++	0.940	0.893	0.850	22

against the original (ra1) as well as entailed reference alignment (ra2). As the Table 3 shows, for both evaluations we achieved F-Measure values better than the two Baselines results (edna, StringEquiv).

Table 3. Results for Conference track.

System	RA1 Reference			RA2 Reference		
	P	R	F	P	R	F
XMap++	0.87	0.49	0.63	0.82	0.44	0.57

2.4 Multifarm

This track is based on the translation of the OntoFarm collection of ontologies into 9 different languages. XMap ++'s results are showed in the Table 4.

Table 4. Results for Multifarm track.

System	Different ontologies			Same ontologies		
	P	F	R	P	F	R
XMap++	0.31	0.35	0.43	0.76	0.50	0.40

2.5 Library

The library track involves the matching of the STW thesaurus (6,575 classes) and the Soz thesaurus (8,376 classes). Both of these thesauri provide vocabulary for economic and social sciences. The results are depicted in table 5; our tools achieved a good recall of ≈88%, and the precision was low (50%). XMap++ requires ≈ 3 hr and 30 min, it is mainly due to the long times required for looking up concepts in Bing Translator when it attempts to translate all the German labels to English labels.

Table 5. Results for Library track.

System	Precision	Recall	F-Measure	Time(s)
XMap++	0.508	0.885	0.646	12652

2.6 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 6 sub-tasks corresponding to different sizes of input ontologies (small fragment and whole ontology for FMA and NCI and small and large fragments for SNOMED CT). The results obtained by XMap++ are depicted on Table 6. In general we can conclude

Table 6. Results for the Large BioMed track.

Test set	Precision	Recall	F-Measure	Time(s)
Small FMA-NCI	0.932	0.848	0.888	17
Whole FMA-NCI	0.835	0.745	0.787	144
Small FMA-SNOMED	0.858	0.737	0.793	35
Whole FMA- Large SNOMED	0.558	0.633	0.593	390
Small SNOMED-NCI	0.849	0.665	0.746	182
Whole NCI- Large SNOMED	0.843	0.584	0.690	490

that Xmap++ achieved a good precision and fair recall value. The fair recall value can be explained by the fact that WordNet does not contain definitions of highly technical medical terms, resulting in the system being unable to match entities that are not located in the WordNet database. Using a different linguistic ontology should alleviate this problem, or ideally the system should automatically select the most appropriate linguistic ontology for this task.

2.7 Ontology Alignment for Query Answering

The objective of this test is to verify the ability of the generated alignments to answer a set of queries in an ontology-based data access scenario where several ontologies exist. The table 7 shows the F-measure results for the whole set of queries. XMap++ was one of the four matchers whose alignments allowed to answer all the queries of the evaluation.

Table 7. Results for Ontology Alignment for Query Answering.

System	RA1 Reference			RAR1 Reference		
	P	R	F	P	R	F
XMap++	0.556	0.487	0.505	0.554	0.487	0.505

3 General comments

3.1 Comments on the results

This is the second time that we participate in the OAEI campaign. While we participated with two configurations of our system to the 2013 edition of the campaign, respectively with XMapGen and XMapSig, this year a unique version has been submitted. Several changes have been introduced. The official results of OAEI 2014 show that XMap++ is competitive with other well-known ontology matching systems in all OAEI tracks, especially in Library track it got the highest recall of all attended systems. The current version of XMap++ has shown a significant improvement both in terms of matching quality and runtime. Additionally, to tackle the large ontology matching problem we improved the runtime of the algorithm using a divide-and-conquer approach that can partition the execution of the matchers into small threads was improved and joins their results after each similarity calculation.

3.2 Discussions on the way to improve the proposed system

Some probable approaches to improving our tools are listed as follows:

1. Take comments and Instance information of ontology into account, especially when the name of a concept is meaningless;
2. Using the UMLS Meta-thesaurus to have high recall when aligning ontologies from the biomedical science domain;
3. Pre-compiling a local dictionary in order to avoid multiple accesses to the Microsoft Translator within the matching process.

3.3 Comments on the OAEI 2013 procedure

As a second participation, we found the OAEI procedure very convenient and the organizers very supportive. The use of Seals allows objective assessments. The OAEI test cases are various, and this leads to comparison on different levels of difficulty, which is very interesting. We found that SEALS platform is a very valuable tool to compare the performance of our system with the others.

4 Conclusion

We have briefly described our fully automate ontology matching system XMap++ and presented the results achieved during the 2014 edition of the OAEI campaign. The obtained results showed that XMap++ is able to efficiently and effectively match ontologies of different size. In future we want to participate in more tracks. Our ontology matching system presents some limitations. We intend to use the UMLS resource for better discarding incorrect mappings for life sciences related ontologies.

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.: XMapGen and XMapSiG results for OAEI 2013. In Proceedings of the 8th International Workshop on Ontology Matching co-located with the 12th International Semantic Web Conference (ISWC 2013), October 21, pp. 203–210. CEUR-WS.org, Sydney, Australia (2013)
3. 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)
4. Djeddi, W., Khadir, M.T.: Introducing artificial neural network in ontologies alignment process. In the Journal Control and Cybernetics, Vol. 41, No. 4, pp.743-759 (2012)
5. Djeddi, W., Khadir, M.T. : A dynamic multistrategy ontology alignment framework based on semantic relationships using WordNet. In Proc of the 3rd International Conference on Computer Science and its Applications (CIIA´11), 13−15 December, Saida, Algeria, pp.149-154 (2011)
6. Djeddi, W., Khadir, M.T.: XMAP: a novel structural approach for alignment of OWL-full ontologies. In Proc. of the International Conference on Machine and Web Intelligence (ICMWI), pp.347-352 (2010)
7. Fellbaum, C. : WordNet: An Electronic Lexical Database, MIT Press, Cambridge, MA (1998)
8. Gross, A., Hartung, M., Kirsten, T. and Rahm, E. : On matching large life science ontologies in parallel. In , in Lambrix, P. and Kemp, G.J.L. (Eds), DILS, Springer, pp.35-49 (2010)

Evaluation of String Normalisation Modules for String-based Biomedical Vocabularies Alignment with AnAGram

Anique van Berne, A.vanBerne@Elsevier.com Elsevier BV

Veronique Malaisé V.Malaise@Elsevier.com Elsevier BV

Abstract: We evaluate the precision and recall of the different normalization modules of AnAGram: a modular string-based vocabulary alignment tool we built for biomedical vocabularies. The main feature of AnAGram is a targeted transformation using a dictionary of adjective/noun correspondences, which gives interesting results. We find that the classic Porter stemming algorithm needs adaption to the biomedical domain in order to produce quality results.

1. Introduction: AnAGram and Related Work

This paper stems from a product interoperability effort in the biomedical domain through taxonomy alignment. Though requiring a generic tool, each individual alignment requires specific conditions to be optimal, due to lexical idiosyncrasies. AnAGram is constructed as a modular, step-wise, string-based alignment tool (as string-based tools perform well on the anatomical datasets of the OAEI campaign[1]).

AnAGram is built for a local system[2], using hash-table lookup for performance. Matching is modular: a user selects one or multiple modules for processing the source taxonomy. The alignment stops at the first match in the target taxonomy. The modules are ordered to produce results of increasing distance from the original string (similar to a confidence value) and include: exact match; stop word removal (using an independent fine-tuned list); re-ordering (sorting tokens alphabetically for multi-word terms match); stemming (with Porter stemmer[3]); normalization (of non-alpha-numeric characters); substitution (replacing adjective/noun from our substitution dictionary).

The modules correspond to the list by Cheatham and Hitzler[4] of syntactic linguistic processes used by at least one alignment tool in the Ontology Alignment Evaluation Initiative (OAEI)[5]. Chua and Kim's[6] approach is closest to AnAGram, using WordNet[7] for building adjective/noun pairs to improve their matches, where ours is built on the biomedical reference Dorland's (creating a larger substitution dictionary).

[1] http://oaei.ontologymatching.org/2013/anatomy/index.html

[2] Dell™ Precision™ T7500, 2x Intel® Xeon® CPU E5620 2.4 GHz processors, 64 GB RAM. Software: Windows 7 Professional 64 bit, Service Pack 1; Perl v5.16.3

[3] http://tartarus.org/martin/PorterStemmer/

[4] http://disi.unitn.it/~p2p/RelatedWork/Matching/strings-iswc13.pdf

[5] http://oaei.ontologymatching.org/2014/

[6] http://www.ncbi.nlm.nih.gov/pubmed/22155335

[7] http://wordnet.princeton.edu/

2. Evaluations and conclusion

As a test case, we align EMMeT[8] to Dorland's (32nd edition). We evaluate a random sample of non-exact alignments (100), comparing them with a baseline Jaro-Winkler (JW) matching approach. AnAGram gives more correct results and JW finds more related matches (Table 1- top two lines, and Figure 1).

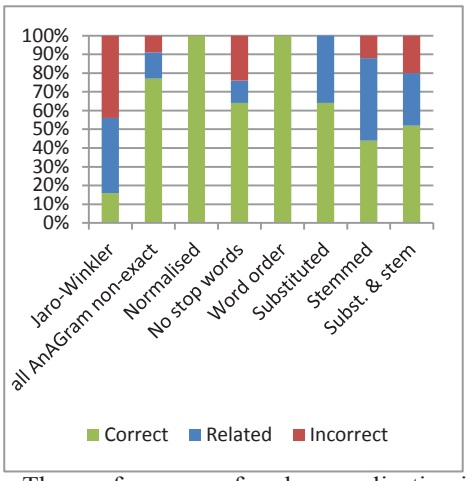


Figure 1 - Quality of matches returned by AnAGram's modules.

Preferred labels	C	R	I
Jaro-Winkler	16	40	44
AnAGram non-exact	77	14	9
Normalised	25	0	0
No stop words	16	3	6
Word order	25	0	0
Substituted	16	9	0
Stemmed	11	11	3
Subst. & stem	13	7	5

Table 1 – Results for AnAGram's modules. (C: correct; R: related; I: incorrect)

The performance of each normalization is evaluated using 25 random results for each of AnAGram's modules separately[9] (Table 1- bottom, Figure 1). Normalization does very well (100% correct results). Removal of stop words causes some errors and related matches (stop words can be meaningful like *A* for *hepatitis A*). Word order rearranging ranks second: it does not often change the meaning of the term. Substitution performs reasonably well: most of the non-correct results are related matches. Stemming gives the poorest results, with false positives due to nouns/verbs stemmed to the same root, such as *cilitated/ciliate*. The substituted-and-stemmed matches have a result similar to the stemmed results. Still, even the worst results from any AnAGram module are better than the overall results of the non-exact matches from the JW algorithm. One reason for this can be that JW does not stop the alignment at the best match, but delivers everything that satisfies the threshold.

Not all modules account for an equal portion of the non-exact results. The normalization module delivers around 70% of matches, stemming accounts for 15 to 20% and the other modules account for 2% to 4% of the matches each.

AnAGram's results are good compared to the performance of string-based methods in the OAEI large biomedical vocabularies alignment[10]. We will work on the Stemming algorithm, on improving our stop words list and substitution dictionary, and on adding an optimized version of the JW algorithm, thus benefitting from additional related matches where no previous match was found.

[8] Version 3.2, from December 2013

[9] Some modules use previous transformation results.

[10] http://oaei.ontologymatching.org/2013/largebio/index.html

Building reference alignments for compound matching of multiple ontologies using OBO cross-products

Catia Pesquita[1], Michelle Cheatham [2], Daniel Faria[1], Joana Barros [1], Emanuel Santos[1], and Francisco M. Couto[1]

[1]Departamento de Informática, Faculdade de Ciências, Universidade de Lisboa, Portugal
[2] DaSe Laboratory, Wright State University, Dayton OH 45435, USA
cpesquita@di.fc.ul.pt

Abstract. Existing ontology matching techniques are limited to matching two ontologies, but we argue that producing 'compound' alignments, involving more than two ontologies, would be useful to support a next generation of semantic technologies. To foster the development of new techniques in this area, we have investigated the suitability of exploring OBO cross-products to derive ternary compound alignments that can be used as a benchmark. We were able to establish seven such reference alignments with over 100 mappings each, between ten biomedical ontologies. Preliminary experiments revealed that the increase in matching space and the inherently more difficult-to-compute ternary mapping pose interesting difficulties to compound ontology matching.

Introduction. Both the 'classical' and 'complex' (e.g., [1–3]) ontology matching approaches focus on discovering mappings between two ontologies. We argue that it would be useful for the developers of ontology alignment systems to develop new techniques and tools for identifying 'compound matches', i.e. matches between class or property expressions involving more than two ontologies. The simplest of these mappings would correspond to an equivalence mapping between a class A of one ontology and an expression relating classes B and C of two other ontologies, constituting a ternary relationship. We investigate the suitability of exploring OBO cross-products to create ternary compound alignments between ontologies which can function as a gold-standard to support the evaluation of novel matching methods for compound alignment.

Approach. We consider that a ternary compound alignment is a set of correspondences (mappings) between classes from a source ontology O_s and class expressions obtained by combining two other classes each belonging to a different target ontology O_{t1} and O_{t2}. We define a ternary compound mapping as a tuple $< X, Y, Z, R, M >$, where X, Y and Z are classes from three distinct ontologies, R is a relation established between Y and Z to generate a class expression that is mapped to X via a mapping relation M. Some of the logical definitions contained in OBO cross-products correspond to this type of mapping, for instance,

the class HP:0000337 labeled *broad forehead* is equivalent to an axiom obtained by relating the classes PATO:0000600 (*increased width*) and FMA:63864 (*forehead*) via an intersection qualified by an *inheres_in* relation. We analyzed the resources available at obofoundry.org [1] and identified seven cross-products collections each with at least 100 definitions corresponding to ternary compound mappings:

Source Ontology	Target Ontologies		Size
MP	PATO	UBERON	1725
HP	PATO	FMA	1519
MP	PATO	CL	407
WBPhenotype	PATO	GO	369
MP	PATO	GO	354
FYPO	PATO	GO	285
MP	PATO	NBO	100

To create the alignments based on the cross-products collections we used EDOAL [4], since it allows the construction of entities from other entities using algebraic operators. To represent *intersection_of* we employed a class expression with the *and* operator.

Experiments. In ternary ontology matching, the search space is cubic, so matching even relatively small ontologies can pose efficiency problems. In a preliminary experiment, we adapted the anchor-based strategy of the AgreementMakerLight system[5] as well as its WordMatcher algorithm to use a modified Jaccard index that penalizes words shared by both target classes. We tested it in the MP-PATO-CL and MP-PATO-NBO alignments, obtaining recall values of 30 and 11% respectively, but precision values below 1%. These results highlight some of the complexity behind compound alignments, even between ontologies that strive to follow the same naming conventions. We posit that to solve these issues, background knowledge or instances would be needed to be able to discriminate between the candidate mappings.

Acknowledgements. This study was funded by the Portuguese FCT through the SOMER project (PTDC/EIA-EIA/119119/2010) and the LASIGE Strategic Project (PEst-OE/EEI/UI0408/2014) and by the National Science Foundation under award 1017225 "III:Small: TROn—Tractable Reasoning with Ontologies."

References

1. Ritze, D., Völker, J., Meilicke, C., Šváb-Zamazal, O.: Linguistic analysis for complex ontology matching. Ontology Matching **1** (2010)
2. Šváb-Zamazal, O., Svátek, V., Iannone, L.: Pattern-based ontology transformation service exploiting oppl and owl-api. In: Knowledge Engineering and Management by the Masses. Springer (2010) 105–119
3. Meilicke, C., Noessner, J., Stuckenschmidt, H.: Towards joint inference for complex ontology matching. AAAI (Late-Breaking Developments) (2013)
4. David, J., Euzenat, J., Scharffe, F., Trojahn dos Santos, C.: The alignment api 4.0. Semantic web **2**(1) (2011) 3–10
5. Faria, D., Pesquita, C., Santos, E., Palmonari, M., Cruz, I.F., Couto, F.M.: The agreementmakerlight ontology matching system. In: On the Move to Meaningful Internet Systems: OTM 2013 Conferences, Springer (2013) 527–541

[1] http://obofoundry.org/index.cgi?show=mappings

A Term-Based Approach for Matching Multilingual Thesauri

Mauro Dragoni[2], Andi Rexha[3], Matteo Casu[1], and Alessio Bosca[1]

[1] Celi s.r.l., Via S.Quintino 31, I-10131, Torino, Italy
[2] FBK–IRST, Trento, Italy
[3] Know-Center Graz, Graz, Austria
dragoni@fbk.eu arexha@know-center.at casu|alessio.bosca@celi.it

Abstract. In this paper, we present a multilingual matching approach aiming at building matches between terms belonging to multilingual thesauri. The approach is presented as a variant of the schema matching problem and present its evaluation on domain-specific use cases by demonstrating the viability of the proposed technique for facing the multilingual thesaurus matching approach.

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 term-based approach for defining mappings between multilingual thesauri. Such an approach 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 term defined in a thesaurus. Let us consider two thesauri: (i) a source thesaurus containing the elements that have to be mapped, and a target thesaurus used as reference for

creating the mappings. The proposed approach has been built by taking inspiration from information retrieval techniques and it exploits the creation of indexes for identifying candidate mappings.

Therefore, the entire approach may be split in two different phases: (i) in the first one, we created the index containing information about the target thesaurus represented in a structured way; while, (ii) in the second phase, we build queries using information contained in the source thesaurus for retrieving a rank representing the candidate mappings that we may define between the two thesauri.

First of all, the two thesauri are considered with two different roles: a source thesaurus that is used as starting point for the creation of the mapping, and a target thesaurus that is considered as ending point of the mapping. It is split in two main phases: in the first one, it operates on the target thesaurus, while in the second one, on the source thesaurus. Firstly, we extract the whole set of labels from the target thesaurus and, after a set of preprocessing activities, each term of the target thesaurus is transformed into a structured representation containing all its multilingual labels and it is 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 thesauri for which gold standards containing the mappings were available. Such thesauri belong to two different domains: three thesauri to the agricultural and environment domain, while the other three to the medical one. The promising results shown in Table 1 demonstrated the effectiveness of the proposed approach.

Mapping Set	**# of Mappings**	**Prec@1**	**Prec@3**	**Prec@5**	**Recall**
Eurovoc → Agrovoc	1297	0.861	0.946	0.978	0.785
Gemet → Agrovoc	1181	0.927	0.973	0.988	0.643
MDR → MeSH	6061	0.746	0.901	0.948	0.799
MDR → SNOMED	19971	0.589	0.793	0.882	0.539
MeSH → SNOMED	26634	0.674	0.853	0.920	0.612

Table 1: Results obtained on the multilingual ontologies used for the Context 1 evaluation.

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. Choi, N., Song, I.Y., Han, H.: A survey on ontology mapping. SIGMOD Rec. **35**(3) (September 2006) 34–41

The importance of cross-lingual information for matching Wikipedia with the Cyc ontology. *

Aleksander Smywiński-Pohl[1,2] and Krzysztof Wróbel[1,2]

[1] Chair in Computational Linguistics, Jagiellonian University,
ul. Łojasiewicza 4, 30-348 Kraków, Poland,
http://www.klk.uj.edu.pl
[2] Faculty of Computer Science, Electronics and Telecommunications,
AGH University of Science and Technology,
al. Mickiewicza 30, Kraków, Poland,
http://www.dsp.agh.edu.pl

Abstract. In this paper we try to answer the question how cross-lingual evidence may improve matching between different classification schemas. We concentrate specifically on the task of mapping between Wikipedia categories and Cyc terms as well as the classification of Wikipedia articles to the Cyc taxonomy and show how this process may be improved by consuming the evidence that is available in different editions of Wikipedia. The results show that the performance of the mapping procedure may be improved from 0.6 to 4.9 percentage points, depending on the number of external Wikipedia editions and the given task.

Keywords: Ontology, ontology mapping, classification, multilingual data, Wikipedia, Cyc

1 Approach

To answer the question how the additional Wikipedia editions influence the performance of the mapping between Wikipedia and Cyc (cf. [2]) we have defined the following tasks: 1) mapping of the Wikipedia categories to Cyc terms; 2) classification of the Wikipedia articles to the Cyc ontology based on the first sentences. In each case the decision of selecting the corresponding Cyc term requires disambiguation of some English expressions against the Cyc ontology. This decision is based on the contextual data that are available for each Wikipedia article and category. Consulting of the supplementary Wikipedia editions extends the context available when making the decision and in general should improve the performance of the corresponding algorithms.

In case of the category mapping (based on the identification of plural head nouns in category names cf. [3]), when an English category is mapped, the corresponding Dutch, German, etc. categories are inspected. Then the parent and

* This work was partly supported by Stuctured Dynamics LLC, partly by the Polish National Center for Research and Development under LIDER/37/69/L-3/11/NCBR/2012 grant and partly by the Faculty of Management and Social Communication, Jagiellonian University in Krakow.

child categories as well as articles of the corresponding categories in the other editions are looked up in a reverse interlingual mapping index and if there is an English Wikipedia page, that was not present in the original context, it is included in the new, extended context. Then a support value used to disambiguate the category is computed against the extended context.

In case of article classification (based on the first sentence parsing, cf. [1]) the supplementary Wikipedia editions provide additional categories for the classified article, that are used to verify the disambiguation decision. The manner of operation is similar to that from the previous task – the corresponding articles in other Wikipedia editions are consulted, their categories are translated back to English and these new categories are included in the extended context.

2 Results

There was a small improvement (F1 increased from 86.8% to 78.4%) in the performance of the category mapping when the English Wikipedia is supported by three other Wikipedias (`de`,`nl`,`sv`). However providing the algorithm with more data from other Wikipedia editions, increased the computation time, but did not further improve the results.

On the other hand the influence of the additional Wikipedia editions in the task of the classification of the articles into the Cyc ontology was much stronger. Not only the additional Wikipedia editions improved the recall, but also the precision. The maximum precision was achieved for 5 and 6 additional Wikipedias (96.6% compared to 95.8% for the sole English Wikipedia). The F1 was the largest for the 8 additional Wikipedias resulting in an increase from 66.2% to 71.1%.

The overall conclusion from the results is that the influence of the supplementary Wikipedias is task dependant and in general the extra time necessary to pre-process the data and the increase of the computation time may not be justified. However the task of articles classification shows also that such supplementary data may be very valuable and may increase both the precision and the recall of the results.

References

1. Gangemi, A., Nuzzolese, A.G., Presutti, V., Draicchio, F., Musetti, A., Ciancarini, P.: Automatic typing of DBpedia entities. In: The Semantic Web–ISWC 2012, pp. 65–81. Springer (2012)
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: Proceedings of the 16th international conference on World Wide Web. pp. 697–706. ACM (2007)

Constructing a Class Hierarchy with Properties by Refining and Aligning Japanese Wikipedia Ontology and Japanese WordNet

Takeshi Morita[1], Susumu Tamagawa[2], and Takahira Yamaguchi[2]

[1] School of Social Informatics, Aoyama Gakuin University, Japan
t_morita@si.aoyama.ac.jp
[2] Graduate School of Science and Technology, Keio University, Japan

Introduction We have proposed learning methods for building a large-scale and high accuracy general ontology called Japanese Wikipedia Ontology (JWO) by extracting the concepts and relationships between concepts from various semi-structured resources in Japanese Wikipedia [3]. However, JWO has problems because it lacks upper classes and appropriate definitions of properties. Thus, the aim of our research was to complement the upper classes in JWO by aligning JWO and Japanese WordNet (JWN) [3] using ontology alignment(OA) techniques. To achieve our aim, we developed tools that help users to refine class-instance relationships, to identify the JWO classes that need to be aligned with JWN synsets, and to align the JWO classes with the JWN synsets via user interaction. We also integrated JWO and JWN by using a domain ontology development environment, DODDLE-OWL [1]. Moreover, we propose a method for building a class hierarchy with defined properties by elevating common properties defined in sibling classes to higher classes in JWO. This research is based on our previous study [2]. The refined JWO and source code of the developed tools can be downloaded via a GitHub repository [4].

Proposed Methods We propose two main types of method: aligning JWO and JWN; and defining the domains of properties based on a consideration of property inheritance. Note that we used the version of JWO from November 2010 and JWN ver. 1.1 in this study. The details of the proposed methods are described in [2]. The procedures used for aligning JWO and JWN are described as follows:

1. Extracting class-instance relationships from the listing pages of Japanese Wikipedia
2. Refining class-instance relationships and identifying alignment target classes
3. Aligning JWO classes and JWN synsets
4. Integrating JWO and JWN using DODDLE-OWL
5. Removing redundant class-instance relationships

[3] http://nlpwww.nict.go.jp/wn-ja/index.en.html
[4] http://t-morita.github.io/JWO_Refinement_Tools/

We used OA techniques to integrate JWO and JWN. OA is usually applied to similar structured domain ontologies. However, the structure of JWO is quite different from that of JWN. Therefore, it is difficult to apply OA techniques using glosses, common instances or properties, and the class hierarchy structure in the two ontologies. Thus, we used methods based on string matching similarity (prefix, suffix, edit distance, and n-gram) as OA techniques to integrate JWO and JWN. The methods we selected are very basic OA techniques and the accuracy of the alignments may be low. Therefore, we developed a tool that supports the alignment of classes in JWO and the synsets in JWN via user interaction. The inputs for the tool are the alignment target classes in JWO. A user can dynamically align the classes in JWO and the synsets in JWN. The user aligned 736 alignment target classes in JWO and sysnsets in JWN using the tool in about 6 hours.

As a result, the number of classes from JWO is 2,787, the number of classes from JWN is 675, the number of instances is 344,934, and the number of class-instance relationships is 444,597.

The procedure of defining domains of properties based on a consideration of property inheritance by refining the definition of the domains of properties in JWO is as follows:

1. Extracting the domains of properties from instance triples and the types of subject resources for the instance triples (If there is an instance triple s-p-o and the type of s is T, the domain of property p is T.)
2. Elevating common properties that are defined in the sibling classes to higher classes in JWO
3. Removing the properties defined in a class that are also defined in super-classes of the class (The properites can be derived using a reasoner, so we regard them as redundant properties and remove them.)

As a result, we extracted 4,357 properties. After elevating the properties and removing the redundant properties that are defined expressly, we reduced the number of domains of properties from 143,500 to 18,678.

References

1. Takeshi Morita, Noriaki Izumi, Naoki Fukuta, Takahira Yamaguchi, *DODDLE-OWL: Interactive Domain Ontology Development with Open Source Software in Java*, IEICE Transactions on Information and Systems, Special Issue on Knowledge-Based Software Engineering Vol.E91-D No. 4 pp. 945-958, 2008.
2. Takeshi Morita, Yuka Sekimoto, Susumu Tamagawa and Takahira Yamaguchi, *Building up a class hierarchy with properties by refining and integrating Japanese Wikipedia Ontology and Japanese WordNet*, Web Intelligence and Agent Systems: An International Journal, Volume 12, Number 2, pp. 211-233, IOS Press, 2014.
3. Susumu Tamagawa, Shinya Sakurai, Takuya Tejima, Takeshi Morita, Noriaki Izumi, and Takahira Yamaguchi, *Learning a Large Scale of Ontology from Japanese Wikipedia*. 2010 IEEE/WIC/ACM International Conference on Web Intelligence, pp. 279-286, 2010.

Partitioning-based Ontology Matching Approaches: A Comparative Analysis

Alsayed Algergawy[1,2], Friederike Klan[1], and Birgitta Konig-Ries[1]

[1] Institute for Computer Science, Friedrich Schiller University of Jena, Germany
[2] Department of Computer Engineering, Tanta University, Egypt
{firstname.lastname@uni-jena.de}

Generic Framework. Ontology matching is the process that takes two or more ontologies to identify semantically corresponding entities across them. As the numbers of developed ontologies as well as the number of entities in each ontology are increasing, traditional approaches to ontology matching fail or are not able to scale. Therefore, there is a growing need for new matching algorithms. A common approach to deal with the large-scale matching problem is the partitioning-based technique [5]. To make these techniques comparable, we propose a generic framework containing the following phases (shown in Fig. 1):

- *Prematch*. This phase aims to prepare input ontologies for matching. It starts by parsing and representing input ontologies as graphs, called *ontology graphs*. The input ontology graphs are then partitioned into a set of sub-ontologies such that entities belonging to one partition are similar (have some common features) while entities from different partitions are dissimilar. The partitioning process may extend from using simple ad hoc rules [2] to clustering algorithms [1,4]. The task now is to determine which partitions of the two sets are sufficiently similar and thus worth to be matched in more detail. The goal is to reduce the matching overhead by avoiding to find correspondences between unrelated partitions.

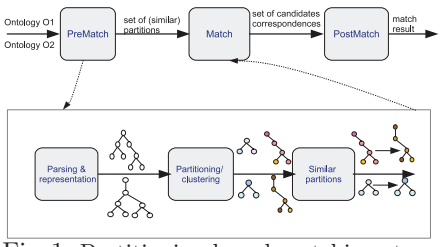


Fig. 1: Partitioning-based matching steps.

- *Match*. Once settling on similar partitions (clusters) of the two ontologies, the next step is to fully match similar clusters to obtain the correspondences between their elements. Each pair of similar partitions represents an individual match task that is independently solved.

- *Postmatch*. Local match results should be merged (combined) to generate the final match result. The Postmatch phase is also concerned with *matching cardinality* and *mapping representation*.

Matching Systems: A Comparison. We aim to present partitioning-based approaches fitting to the algorithmic steps identified above indicating which part of the solution is covered by which prototypes, thereby supporting a comparison of these approaches. We notice that all these approaches use the graph

data structure as the internal data representation. However, they utilize different algorithms to partition the ontology graph. Falcon-AO [4] and the extension of COMA++ [1] employ an agglomerative clustering algorithm, which independently partitions input ontologies. To dependently partition ontologies, TaxoMap [3] uses a co-clustering technique. It is worth noting that some matching approaches first partition the ontology graphs and then determine similar partitions such as COMA++ [1,2] and Falcon-AO [4], while others determine similar partitions during the partitioning process such as TaxoMap [3]. We also observe that to determine similar partitions the matching approaches use different methods extending from exploiting only the partitions' roots, e.g. COMA++ [2], to exploiting the whole partition information, e.g. Falcon-AO [4]. Some other approaches compromise between the two extremes, e.g. the extension of COMA++ [1] exploits entity names to find similar partitions.

From the matching phase point of view, each matching system uses its own matching strategy which exploits linguistic and structural features of ontologies. Some of these systems make use of existing matching strategies, such as TaxoMap (using the Falcon-AO match strategy) and the Unbalanced OM approach utilizing the similarity flooding algorithm. More specifically, this means that these matching systems do not implement matching strategies specific to this kind of matching, however, they utilize off-the-shelf matching strategies.

It is also worth noting that some matching approaches interlink between the last two phases, i.e. they do not focus on getting local match results for each matching task, but directly construct the final match result. Other matching approaches, like COMA++, first consider each match task as a completely independent match task getting its own local results and then merge or combine these local results to get the final match result.

Future Directions. In this paper we introduced a first conceptual comparison of partitioning-based matching approaches. This will be followed up by an experimental evaluation to determine which combination of approaches works best in which circumstances and to identify necessary areas of improvement.

References

1. A. Algergawy, S. Massmann, and E. Rahm. A clustering-based approach for large-scale ontology matching. In *ADBIS*, pages 415–428. Springer, 2011.
2. H. H. Do and E. Rahm. Matching large schemas: Approaches and evaluation. *Information Systems*, 32(6):857–885, 2007.
3. F. Hamdi, B. Safar, C. Reynaud, and H. Zargayouna. Alignment-based partitioning of large-scale ontologies. In *SCI*, volume 292, pages 251–269. 2010.
4. W. Hu, Y. Qu, and G. Cheng. Matching large ontologies: A divide-and-conquer approach. *DKE*, 67:140–160, 2008.
5. E. Rahm. *Schema Matching and Mapping*, chapter Towards Large-scale Schema and Ontology Matching, pages 3–27. 2011.

Towards a Cluster-based Approach for User Participation in Ontology Maching

Vinicius Lopes, Fernanda Baião, Kate Revoredo

Department of Applied Informatics
Federal University of the State of Rio de Janeiro (UNIRIO), Rio de Janeiro, Brazil
{vinicius.lopes,fernanda.baiao,katerevoredo}@uniriotec.br

Abstract. User participation is a promising approach for Ontology Matching; however, determining the most representative pairs of entities is still a challenge. This paper delineates an Ontology Matching approach for user participation employing a clustering algorithm.

Keywords. ontology matching, machine learning, clustering

1 Introduction

Ontology matching focuses on identifying correspondences between entities of two or more ontologies and establishing an alignment as a solution to the heterogeneity problem. Some works in ontology matching apply user participation approaches [2][5], such as selecting and combining similarity measures, tuning parameter values or giving feedback for suggested correspondences. User feedback is considered a promising approach since it requires domain knowledge as opposed to technical knowledge. Due to the difficulty of finding available users, however, it is necessary to minimize user effort by selecting the most representative correspondences. This work delineates an approach to address this issue, in which we apply a clustering algorithm to identify the most representative pairs of entities.

2 A Clustering-based Approach for User Participation

Our proposed approach is composed by 4 steps, which are detailed below.

Select Candidate Correspondences. In this step, a committee is formed to select a subset of candidate correspondences for the user feedback. Given two ontologies O and O´, each committee member m_i is represented by a matrix M_i. Each cell $M_i[x,y]$ is the similarity value (calculated according to a unique or a combination of similarity measures) for the pair (x,y), where x is an entity of O and y is an entity of O´. Since Mi are typically sparse matrices (given that most of the pairs do not match), this step analyzes all matrices and selects pairs with the highest potential for actually being correspondent. A pair (x,y) is selected as a candidate correspondence iff, for every matrix Mi, y is the entity that is most similar to x, and vice-versa.

Select Correspondences for User Feedback. In this step, we apply the algorithm farthest-first [1] as a naïve, yet effective and efficient clustering algorithm for selecting correspondences for user feedback among the candidate correspondences. Each instance to be clustered represents a candidate

correspondence (x, y). The attributes of an instance (x, y) are the similarity values Mi[x][y] of each matrix. The cluster centroids are selected for user feedback and then stored in a repository.

Collect and Propagate User Feedback. The user gives his feedback on the selected pairs (either confirming or rejecting as a real correspondence). The feedbacks are updated in the repository.

Learn the Ontology Alignment and Propagate User Feedback. In this step, a classification algorithm is executed considering the repository of classified correspondences. The Naive Bayes classification algorithm achieved the best results. The bayes rule determines the probability distribution of class *C* for a pair of entities, considering its attributes (similarity measures). The resulting model is used to classify candidate correspondences, returning the label *c* that maximizes the posterior probability to propagating the effect of user feedback for the remaining candidate correspondences, and storing them in the repository.

We executed an initial experiment of the approach on top of the OAEI conference dataset. Reference alignments were used to validate the results and simulate user feedbacks. We considered only equivalence correspondences between classes. The committee included Cosine [4] and WuPalmer [3] similarity measures. We evaluated two values (3 and 6) for the number of clusters, or user feedbacks. In the first run the approach achieved an average precision of 0.68 and an average recall of 0.55. In the second run the approach achieved an average precision of 0.83 and an average recall of 0.58. These results show an increase in the precision of 15% when the number of feedbacks increases. F-measure also increased from 0.58 from 0.67. However, the metrics remained the same (or even decreased) for certain pairs of ontologies, indicating there is a need to further investigate the optimal number of clusters for each case.

3 Conclusion

We introduce an approach for ontology matching with user participation that selects candidate correspondences based on a committee of similarity measures. Promising results were obtained on top of the OAEI conference dataset. Future work will perform further experiments, consider other similarity measures and clustering algorithms (including hierarchical approaches).

References

1. Gonzalez, T. F. Clustering to minimize the maximum intercluster distance. Theoretical Computer Science 38, pp. 293–306 (1985).
2. Cruz, I.F., Stroe, C., Palmonari, M.: Interactive User Feedback in Ontology Matching Using Signature Vectors. In: Kementsietsidis et al (eds.) ICDE. pp. 1321–1324 (2012).
3. Wu, Z., Palmer, M.: Verb Semantics and Lexical Selection. Proc. 32nd annual meeting on Association for Computational Linguistics. pp. 133–138 (1994).
4. Stoilos, G., Stamou, G., Kollias, S.: A String Metric for Ontology Alignment. Semant. Web–ISWC 2005. 3729, 624–637 (2005).
5. Shi, F., Li, J., Tang, J., Xie, G.T., Li, H.: Actively Learning Ontology Matching via User Interaction. In: Bernstein, A. et al (eds.). ISWC. pp. 585–600. Springer (2009).

One Query at a Time: Incremental, Collective Ontology Matching

Thomas Kowark and Hasso Plattner

Hasso Plattner Institute
August-Bebel-Str. 88, 14482 Potsdam, Germany
{firstname.lastname}@hpi.de

1 Introduction

Ontology matching is not an end in itself but a prerequisite for applications like query answering over different data sources. In software repository analysis, for example, queries reflect process metrics that researchers want to investigate. Hence, being able to answer such queries on multiple data sets without having to perform data transformations or manual query rewriting is a desirable objective. Unfortunately, the terminological differences and often complex correspondences between different software repository representations impede a completely automatic matching and necessitate user input to create more comprehensive alignments [4].

Our work is concerned with the question of when and how users should introduce their expertise into the matching process. Similar to other approaches, such as the keyword-based information retrieval tasks, which Ellis et al. [1] use to extract user knowledge about ontology alignments, we integrate the input process into the desired application – query translation between different software repositories. From the way this task is carried out, both simple and complex correspondences are inferred. In future translation tasks, these correspondences are reused to incrementally reduce the number of required user interactions. As users only translates their queries of interest, the overall effort for alignment creation is collectivized. In this poster, we present the general architecture of our system and the rules used for complex alignment extraction.

2 Rule-Based Inference of Ontology Alignments

Our approach assumes a setup were a source and a target repository are described by ontologies $\mathcal{O}_{R1}$ and $\mathcal{O}_{R2}$ containing the respective TBoxes $\mathcal{T}_{R1}$ and $\mathcal{T}_{R2}$, respectively. A query translation thus aims to recreate a query, which was originally issued on $\mathcal{O}_{R1}$, by using concepts from $\mathcal{T}_{R2}$. A graph based abstraction is used to express the queries in a query language independent manner [2]. After a preprocessing step performed an automatic ontology matching, users can transform the remaining unmatched elements of those query graphs using the editor shown in Figure 1. To this end, they select the input element(s) and provide an according output graph. In simple cases, the output graph is structurally similar

to the input graph, and only node and edge labels change. Inference of element correspondences is straightforward and comprises concept equivalence and subsumption. If relabelling does not suffice and the graph structure changes, complex correspondences are inferred. Our system employs a rule set to determine which types of correspondences users provide through their input/output graph transformations. The rules are based on the patterns identified by Ritze et al. [3]. For any subsequent query translations, existing transformations are automatically applied by the system, hence, users only have to provide correspondences for missing elements and the required manual effort gradually decreases.

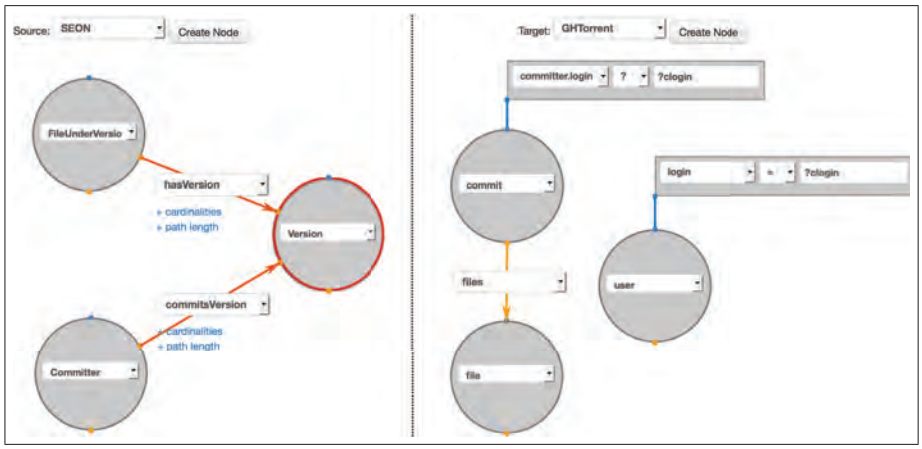


Figure 1. Query graph editor for translating input graph elements (marked red) to an output graph on the right hand side.

References

1. Ellis, J.B., Hassanzadeh, O., Srinivas, K., Ward, M.J.: Collective ontology alignment. In: Proceedings of the Ontology Matching Workshop (2013)
2. Kowark, T., Dobrigkeit, P., Zeier, A.: Towards a shared repository for patterns in virtual team collaboration. In: 5th International Conference on New Trends in Information Science and Service Science (2011)
3. Ritze, D., Meilicke, C., Sváb-Zamazal, O., Stuckenschmidt, H.: A pattern-based ontology matching approach for detecting complex correspondences. In: Shvaiko, P., Euzenat, J., Giunchiglia, F., Stuckenschmidt, H., Noy, N.F., Rosenthal, A. (eds.) OM. CEUR Workshop Proceedings, vol. 551. CEUR-WS.org (2008)
4. 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), http://dx.doi.org/10.1109/TKDE.2011.253

Enabling Semantic Search for EO Products: an Ontology Matching Approach*

M. Karpathiotaki[1], K. Dogani[1], and M. Koubarakis[1]

National and Kapodistrian University of Athens, Greece
{mkarpat,kallirroi,koubarak}@di.uoa.gr

Access to Earth Observation (EO) products remains difficult for end-users. To address this, we developed the Prod-Trees platform[1][2], a semantically enabled search engine for EO products. Users guide their search through a number of ontologies related to EO domain. To facilitate users in finding terms that fit better to their needs, we created mappings between these ontologies. In this paper, we present Pythia, an ontology matching system that utilizes and combines various matching techniques [1,3,4] to create mappings between two ontologies.

Pythia is a combination of a string-based technique utilizing Apache Lucene's features, a language-based technique based on WordNet, and a graph-based technique that uses the structure of the ontology and the mappings produced by the two previous techniques. The system supports SKOS ontologies. Therefore, the mappings are also expressed in SKOS using the defined properties for matching concepts: *skos:exactMatch*, *skos:relatedMatch*, *skos:broadMatch*, and *skos:narrowMatch*. Based on these, we create four different types of mappings.

A **terminological matcher** is responsible for implementing the string- and language-based techniques, both applied on the concepts labels (*skos:prefLabel*, *skos:altLabel* and *skos:hiddenLabel*). The mappings created by this component can either be *skos:exactMatch* or *skos:relatedMatch*.

The **string-based technique** uses Lucene for indexing and searching. With Lucene, one can create documents and add fields of a specific type to these documents. When searching the documents, the user can specify which field he wants to search. Taking advantage of Lucene capabilities, the terminological matcher indexes the target ontology. A new document is created for each concept and each available property of the concept is added as a new field. String normalization functions are applied to the field and unnecessary stop words are removed.

When searching for concepts similar to concept A (from the source ontology), the prefLabel, altLabel, and hiddenLabel fields of the indexed ontology are searched using the prefLabel of concept A. The search results fetched back, are ranked according to the string similarity of the compared strings (e.g., *skos:prefLabel* of A and the prefLabel field of a document). This is feasible due to the string similarity functions implemented in Lucene. Also, since each field is indexed, only the index of the specified field is searched, and not all the concepts.

Lucene returns multiple related results. If the two strings are the same, a *skos:exactMatch* is created between A and the corresponding concept from the

* This work was supported by the Prod-Trees project funded by ESA ESRIN.

[1] A video demonstrating the functionalities of the Prod-Trees platform is available at http://bit.ly/ProdTreesPlatform.

target ontology. Otherwise, and only if one string is a substring of the other (e.g., "Elevation" and "Digital Elevation Model"), a *skos:relatedMatch* is created.

The **language-based technique** uses WordNet, a lexical database for English. The technique is optional and can be bypassed, as it adds noise to the results. Putting WordNet to use, a new field, called *relLabel*, is created in the Lucene document of each concept. *relLabel* enhances each concept's labels, by adding synonyms and other related words found in WordNet. During the search, the relLabel fields of the documents are searched, and if a similarity is discovered, a *skos:relatedMatch* relation is created between the corresponding concepts.

In case there are concepts from the source ontology with no *skos:exactMatch* mappings, a **structural matcher** is invoked. This component implements a graph-based technique creating either *skos:narrowMatch* or *skos:broadMatch* mappings. Taking as input a concept A from the source ontology, the matcher finds all the broaders and narrowers of A. Afterwards, it checks whether a *skos:exactMatch* was created by the terminological matcher for one of these concepts. If it did, then a new mapping can be derived. For example, if a *skos:exactMatch* exists between concept B (which is a broader of A) and concept B'(from the target ontology), then it can be derived that B' will be a *skos:broadMatch* of A. Similarly, we can create a *skos:narrowMatch*.

The matcher also checks whether the concepts B and N hold *skos:narrowMatch* or *skos:broadMatch* relations with concepts from the target ontology. If a *skos:broadMatch* exists between B and a concept B", then it is safe to conclude that B" will also be a *skos:broadMatch* of A. This means that when a *skos:broadMatch* exists between a concept B from the source ontology and a concept B" from the target ontology, then this relation can be propagated to concept's B narrowers. Similarly, a skos:narrowMatch between a concept N and a concept N", can be propagated to concept's N broaders. In any other case, no mappings can be derived. When all the concepts are examined, if new mappings were created by the structural matcher, the described process is repeated. Otherwise, Pythia proceeds with the exportation of the mappings to RDF.

Despite the simplicity of the techniques, the results are quite satisfying. Especially, the performance of the language-based technique, which allows tuning WordNet. By stating the types of relations WordNet discovers for a given word, it gives control over the percentage of valid mappings. A higher degree of trust for the final results can be gained with extensions such as a user-evalutation process and the use of domain-specific vocabularies coupled with Wordnet.

References

1. Euzenat, J., Shvaiko, P.: Ontology Matching (2007)
2. Karpathiotaki, M., et. al.: Prod-Trees: Semantic Search for Earth Observation Products. In: ESWC. LNCS, Springer (2014)
3. Nagy, M., Vargas-Vera, M.: Towards an Automatic Semantic Data Integration: Multi-agent Framework Approach. In: Semantic Web. InTech (2010)
4. Pirro, G., Talia, D.: An approach to Ontology Mapping based on the Lucene search engine library. DEXA '07 (2007)