Ontology Matching

OM-2018

Proceedings of the ISWC Workshop

Introduction

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

The workshop had three goals:

- To bring together leaders from *academia*, *industry* and *user institutions* to assess how academic advances are addressing real-world requirements. The workshop strives to improve academic awareness of industrial and final user needs, and therefore, direct research towards those needs. Simultaneously, the workshop serves to inform industry and user representatives about existing research efforts that may meet their requirements. The workshop also investigated how the ontology matching technology is going to evolve, especially with respect to data interlinking, process mapping and web table matching tasks.
- To conduct an extensive and rigorous evaluation of ontology matching and instance matching (link discovery) approaches through the OAEI (Ontology Alignment Evaluation Initiative) 2018 campaign[2].
- 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, or the emerging process matching task.

The program committee selected 5 submissions for oral presentation, treated as long technical papers in the proceedings, and 9 submissions for poster presentation, out of which 3 are treated as short technical papers and 6 as posters in the proceedings. 19 matching systems participated in this year's OAEI campaign. Further information about the Ontology Matching workshop can be found at: `http://om2018.ontologymatching.org/`.

---

[1] `http://www.ontologymatching.org/`

[2] `http://oaei.ontologymatching.org/2018`

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[3] initiative of the European Network of the Living Labs[4] at Trentino Digitale[5], the EU SEALS (Semantic Evaluation at Large Scale) project[6], the EU HOBBIT (Holistic Benchmarking of Big Linked Data) project[7], the Pistoia Alliance Ontologies Mapping project[8] and IBM Research[9].

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

December 2018

---

[3] `http://www.taslab.eu`
[4] `http://www.openlivinglabs.eu`
[5] `http://www.trentinodigitale.it`
[6] `http://www.seals-project.eu`
[7] `https://project-hobbit.eu/challenges/om2018/`
[8] `http://www.pistoiaalliance.org/projects/ontologies-mapping/`
[9] `http://research.ibm.com/labs/watson/`

Organization

Organizing Committee

Pavel Shvaiko, Trentino Digitale SpA, Italy
Jérôme Euzenat, INRIA & University Grenoble Alpes, France
Ernesto Jiménez-Ruiz, The Alan Turing Institute, UK & University of Oslo, Norway
Michelle Cheatham, Wright State University, USA
Oktie Hassanzadeh, IBM Research, USA

Program Committee

Alsayed Algergawy, Jena University, Germany
Manuel Atencia, University Grenoble Alpes & INRIA, France
Zohra Bellahsene, LIRMM, France
Marco Combetto, Trentino Digitale, Italy
Valerie Cross, Miami University, USA
Jérôme David, University Grenoble Alpes & INRIA, France
Gayo Diallo, University of Bordeaux, France
Warith Eddine Djeddi, LIPAH & LABGED, Tunisia
Zlatan Dragisic, private individual, Sweden
Daniel Faria, Instituto Gulbenkian de Ciência, Portugal
Alfio Ferrara, University of Milan, Italy
Wei Hu, Nanjing University, China
Ryutaro Ichise, National Institute of Informatics, Japan
Antoine Isaac, Vrije Universiteit Amsterdam & Europeana, Netherlands
Marouen Kachroudi, Université de Tunis El Manar, Tunis
Patrick Lambrix, Linköpings Universitet, Sweden
Vincenzo Maltese, University of Trento, Italy
Fiona McNeill, University of Edinburgh, UK
Christian Meilicke, University of Mannheim, Germany
Peter Mork, MITRE, USA
Andriy Nikolov, Metaphacts GmbH, Germany
Axel Ngonga, University of Paderborn, Germany
Catia Pesquita, University of Lisbon, Portugal
Umberto Straccia, ISTI-C.N.R., Italy
Cássia Trojahn, IRIT, France
Ludger van Elst, DFKI, Germany
Ondřej Zamazal, Prague University of Economics, Czech Republic

Table of Contents

Long Technical Papers

Short Technical Papers

---

[10] Paper published in Emerging Topics in Semantic Technologies. ISWC 2018 Satellite Events. E. Demidova, A.J. Zaveri, E. Simperl (Eds.), ISBN: 978-3-89838-736-1, 2018, AKA Verlag Berlin

OAEI Papers

Posters

Matching Domain and Top-level ontologies exploring Word Sense Disambiguation and Word Embedding

Daniela Schmidt*, Rafael Basso*, Cassia Trojahn†, Renata Vieira*

*Pontifical Catholic University of Rio Grande do Sul (Brazil)
{daniela.schmidt, rafael.basso}@acad.pucrs.br, renata.vieira@pucrs.br
†Université de Toulouse 2 & IRIT (France)
cassia.trojahn@irit.fr

Abstract. Top-level ontologies play an important role in the construction and integration of domain ontologies, providing a well-founded reference model that can be shared across domains. While most efforts in ontology matching have been particularly dedicated to domain ontologies, the problem of matching domain and top-level ontologies has been addressed to a lesser extent. This is a challenging task in the field, specially due to the different levels of abstraction of these ontologies. This paper addresses this problem by proposing an approach that relies on existing alignments between WordNet and top-level ontologies. Our approach explores word sense disambiguation and word embedding models. We evaluate our approach in the task of matching DOLCE and SUMO top-level ontologies to ontologies from three different domains.

1 Introduction

Guarino [7] classifies ontologies according to their "level of generality", in particular: *(i) top-level ontologies* describe very general concepts (e.g., space, time, object, etc.), which are independent of a particular problem or domain. These ontologies, also named upper or foundational ontologies [33], are equipped with a rich axiomatic layer; *(ii) domain ontologies* that describe the entities and other information related to a generic domain (e.g., biology or aeronautic). While the rich semantics and formalization of top-level ontologies are important requirements for ontology design [18], they act as well as semantic bridges supporting very broad semantic interoperability between ontologies [15,16]. In that sense, they play a key role in ontology matching.

Whereas the area of ontology matching [3] has developed fully in the last decades, matching ontologies from different levels of abstraction as domain and top-level ontologies is still an early tackled challenge. This is a complex task, even manually, that requires the deep identification of the semantic context of concepts and, in particular, the identification of subsumption relations. The latter is largely neglected by most state-of-the-art matchers. The main problem of matching top-level and domain ontologies using these matching systems is that, despite the variety of approaches, most of them typically rely on string-based techniques as an initial estimate of the likelihood that two elements refer to the same real world phenomenon, hence the found correspondences represent equivalences with concepts that are equally or similarly written. However, in many cases, this correspondence is wrong [32]. In fact, when having different levels of abstraction it might be the case that the matching process is rather capable of identify subsumption correspondences than equivalence, since the top ontology has concepts at a higher level. Approaches dealing with this task are mostly based on manual matching [1,18].

This paper proposes an approach to match domain and top-level ontologies that exploits existing alignments between top-level ontologies and WordNet [20]. These alignments act as bridges for aligning domain and top-level ontologies. The notion of *context* of concepts is used for disambiguating the senses that better express the meaning of domain ontology entities in this external resource. Contexts are constructed from the available terminological information about a domain ontology entity (e.g., entity naming, annotations, and information on the neighbours of entities[1]). Here, we exploit two similarity measures for synset disambiguation : (1) an adaptation of the Lesk measure [13] and (2) word embeddings [19]. Once the domain synset has been selected, we exploit the relation between this synset and a top-level concept via existing alignments between WordNet and the top-level ontologies. Most strategies we apply here, in particular indirect matching [9,8,39], WordNet-based matching [14,38], the classical notion of context [37,30,2] and word-sense disambiguation [21], have been already exploited in different ways in the field. However, we argue that the novelty of our approach relies on their combination, which remains unexplored in the specific task of matching top-level and domain ontologies. The use of word embedding for the matching task is, however, less studied [40,36]. Here, we focus on DOLCE and SUMO top-level ontologies and on their alignments to WordNet [6,23]. This choice is motivated by the fact that they are the most used top-level ontologies and serve as a reference model for the modelling and integration of ontologies [24]. We align them to ontologies from three domains (SSN[2], CORA [28], and OAEI Conference ontologies[3]).

The main contributions of our paper can be summarised as follows : (i) to the best of our knowledge, our approach is the first attempt to automatically match domain and top-level ontologies; (ii) we provide an evaluation of our approach and compare how state-of-the-art matching results can be improved by exploiting existing alignments between WordNet and top-level ontologies; and (iii) our results may form a baseline for an OAEI task since there is no current track involving this kind of challenge.

The rest of the paper is organised as follows. §2 introduces top-level ontologies, WordNet and existing alignments to WordNet. §3 discusses the main related work. §4 presents our matching approach. §5 presents the experiments and discusses the results. Finally, §6 concludes the paper and presents future work.

2 Background

2.1 Top-level ontologies

A top-level ontology is a high-level and domain independent ontology. The concepts expressed are intended to be basic and universal to ensure generality and expressiveness for a wide range of domains. It is often characterized as representing common

---

[1] Here, we do not exploit restrictions and other axioms (e.g., disjointness)

[2] https://www.w3.org/TR/vocab-ssn/

[3] http://oaei.ontologymatching.org/2017/conference/index.html

sense concepts and is limited to concepts which are meta, generic, abstract and philosophical. Several top-level ontologies have been proposed in the literature. The reader can refer to [15] for a review of them. Here, we briefly introduce DOLCE and SUMO, which are further used in our experiments. DOLCE [5,17] (Descriptive Ontology for Linguistic and Cognitive Engineering) was designed to include the most reusable and widely applicable upper-level categories, rigorous in terms of axiomatization and extensively researched and documented. It is an ontology of particulars which has four top-level concepts: *endurant*, *perdurant*, *quality*, and *abstract*. *Endurants* represent objects or substances while *perdurants* correspond to events or processes. The main relation between *endurants* and *perdurants* is that of participation, e.g., a person which is an *endurant*, may participate in a discussion, which is a *perdurant*. *Qualities* can be seen as the basic entities that we can perceive or measure, e.g., shapes, colors, sizes, etc. *Abstracts* do not have spatial or temporal qualities, and they are not qualities themselves. DOLCE has many variations, such as DOLCE-Lite [6], which is an OWL-DL fragment of DOLCE. DOLCE-Lite has been extended in modules for representing information, communication, plans, and with some domain information for representing e.g. legal, biomedical notions. The combination of DOLCE-Lite and the mentioned additional modules is called DOLCE-Lite-Plus[4].

SUMO [22] (Suggested Upper Merged Ontology) provides definitions for general-purpose terms and acts as a foundation for more specific ontologies. It is being used for research and applications in search, linguistics and reasoning. It is an ontology of particulars and universals which has two top-level concepts: *physical* and *abstract*. *Physical* represent an entity that has a location in space-time. An *abstract* can be said to exist in the same sense as mathematical objects such as sets and relations, but they cannot exist at a particular place and time without some physical encoding or embodiment.

2.2 WordNet and its alignments to top-level ontologies

WordNet [20] is a general-purpose large lexical database of English frequently adopted as an external resource in automatic ontology matching between domain ontologies [38,37,30]. In the following, we discuss its alignments to top-level ontologies.

DOLCE to WordNet alignment (OntoWordNet) Gangemi et al. [6] developed the OntoWordNet, a resource which expresses the alignment between WordNet 1.6 version and DOLCE-Lite-Plus. The authors assume that the hyponymy relation could be aligned to the subsumption relation and the synset notion could be aligned to the notion of concept. In OntoWordNet, the named concepts were normalized to obtain one distinct name for each synset. Hence, if a synset had a unique noun phrase, it is used as a concept name (*e.g.* Document__Written_Document__Papers). If the noun phrase was polysemous, the concept was numbered (*e.g.* Writting_1, Writting_2). Figure 1 presents a fragment of WordNet synsets (as concepts) linked to DOLCE-Lite-Plus concepts. The first-level concepts (in lower case) correspond to a DOLCE-Lite-Plus concept. The upper case concepts represent WordNet synsets. Each concept in OntoWordNet is associated to an annotation containing the corresponding gloss of the synset in WordNet.

---

[4] http://www.loa.istc.cnr.it/old/ontologies/DLP_397.owl

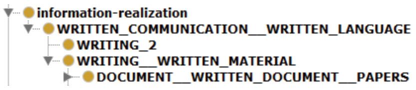


Fig. 1. Example of WordNet synsets linked to DOLCE.

SUMO to WordNet alignment Niles and Pease [23] construct an alignment between SUMO and WordNet 1.6 (a more recent release considers WordNet 3.0). For each identified correspondence, the synset of WordNet is augmented with three information : *(i)* a prefix (&%) that indicates that the term is taken from SUMO; *(ii)* the SUMO concept; and *(iii)* a suffix indicating the kind of relation. The suffix '=' indicates that the correspondence relation is synonymy. '+' indicates that the concept is a hypernym of the associated synset. The instantiation relation is indicated by the suffix '@'. An example of the structure of a correspondence representing a synonymy relation can be seen below. In the example, *"02761392 06 n 03 automaton 0 robot 0 golem"* corresponds to the synset. The gloss is defined as *"a mechanism that can move automatically"*, the prefix *"&%"* indicates that the term is taken from SUMO. *"Device"* corresponds to the SUMO concept and the signal *"+"* is the suffix indicating the hyponymy relation.

02761392 06 n 03 automaton 0 robot 0 golem — a mechanism that can move automatically
&%Device+

There are other efforts that provide alignments of WordNet to top-level ontologies (as Cyc and BFO). The reader can be refer to [29,34] for details.

3 Related work

This section discusses works on aligning domain and top-level ontologies, Wordnet as background knowledge in the matching task, and word embeddings.

Domain and top-level ontology matching. We see a growing importance of aligning domain and top-level ontologies. In [26], correspondences between DBPedia ontology and DOLCE-Zero [4] are used to identify inconsistent statements in DBPedia. In that sense, in [18], a domain ontology describing web services (OWL-S) is manually aligned to DOLCE-Lite-Plus, in order to overcome conceptual ambiguity, poor axiomatization, loose design and narrow scope of the domain ontology. In [35] an alignment between an upper ontology (BFO) and a biomedical ontology (GO) is used for filtering out correspondences at domain level that relate two different kinds of ontology entities. Analysing the impact of using top ontologies as semantic bridges has been done in [16], where a set of algorithms exploiting such bridges are applied and the circumstances under which upper ontologies improve matching approaches are studied. A close approach to ours in terms of data set has been proposed by [25], where OAEI Conference ontologies were manually aligned to UFO, adopting a set of patterns grounded by UFO ontology. There are also works concerning alignment between different top ontologies. In [10,11], the ROMULUS repository aims at improving semantic interoperability between foundational ontologies (DOLCE, BFO and GFO), which are aligned with each other in a semi-automatic way using available matching tools, whose results have been manually evaluated. While these proposals mainly generate manual alignments between top level and domain ontologies, here we propose an approach to automatise this task. A preliminar study is presented in [31].

WordNet as a resource to ontology matching and contexts. Background knowledge from resources such as WordNet has been largely exploited in ontology matching. In [12], a lexical measure considers aggregating sets including names of ontology entities and WordNet synset's words (including hypernyms and meronyms relations). In [38], a set of twelve element-level matchers using WordNet as background knowledge is proposed. The use of WordNet is frequently coupled with the notion of context. In [30], virtual documents (context) represent the meaning of ontology entities and WordNet entries and entities are coupled according to their document similarities. The notion of context has also been exploited in [37], where *semantic description documents* refer to the information about concept hierarchies, related properties and instances, or in [2] where a bag of words describing a concept is exploited within a mining approach. On the other hand, the use of context is very common in the Word Sense Disambiguation, which can be carried out using a diversity of approaches [21]. Here, we adapt the [13] Word Sense Disambiguation to the task of synset disambiguation.

Word embeddings in ontology matching. Word embedding has been largely adopted in several tasks of NLP [19]. It is an umbrella name for a set of NLP language modelling and feature learning techniques which represent words as vectors in a semantic space. Models are trained to produce a vector space and reconstruct the linguistic contexts of words. Each unique word in the corpus is assigned a corresponding vector in the space. Word vectors are positioned in the vector space such that words that share common contexts in the corpus are located in close proximity to one another in the space. The similarity between words is calculated using functions as the cosine similarity, Euclidean distance. Such approach represents an alternative to WordNet similarities, which may fail due to the low WordNet coverage of specific domains. To the best of our knowledge, few works have exploited word embeddings in ontology matching [40,36]. In [40], a hybrid approach combines word embeddings and lexical similarities. The performance of edit distance, WordNet, Latent Semantic Analysis (LSA), word embeddings (using Wikipedia Word2Vec trained model) and the hybrid method were compared, showing that the performance of the hybrid method outperforms the others. In [36], the approach relies on word-to-word similarities exploiting the GloVe model. The hypothesis is that two entities can be matched based on the words in their names using the word-to-word similarity provided by the model. Close to [40], but for a different task, we combine WordNet and word embeddings.

4 Our approach

Our matching approach has two main steps. The first step disambiguates the domain concept, selecting the most appropriated WordNet synset; and the second matches the domain concept to the top-level concept via existing correspondences between WordNet and the top-level ontologies, as detailed below.

4.1 Synset disambiguation

In order to select the synset that better expresses the meaning behind the ontology concept, we adopt the notion of *context*. A *context* is constructed from all information available about an ontology entity, including entity naming (ID), annotation properties (usually labels and comments) and information on the neighbours (super and sub-concepts). Given $Sup(e)$ and $Sub(e)$, the sets of terms denoting the super-concepts and sub-concepts of the entity e, and $Ann(e)$ the set of terms from its annotations, a naive strategy for building a *context* ($context$) considers these sets as a *bag of words* :

$$context(e) = \{e, w | w \in Sup(e) \cup w \in Sub(e) \cup w \in Ann(e)\}$$

This context is used to find the closer synset using two strategies, as above.

Lesk measure The Lesk measure for word sense disambiguation [13] relies on the calculation of the word overlap between the sense definitions of two or more target words. Given a word w, it identifies the sense of w whose textual definition has the highest overlap with the words in the context of w :

$$score_{Lesk}(S) = |context_{Lesk}(w) \cap gloss(S)|$$

where $context_{Lesk}(w)$ is the bag of all content words in a context window around the target word w. Here, we overlap the $context_e$ with the context of each WordNet synset :

$$context(synset) = \{w | w \in Terms(synset) \cap w \in Gloss(synset)\}$$

where $Terms(synset)$ the set of terms in a *synset* and $Gloss(synset)$ the corresponding set of terms from the gloss (i.e, textual description containing definitions and examples) associated to the synset. We hence retrieve the highest overlap between $context(e)$ and $context(synset)$

$$score'_{Lesk}(e) = |context(e) \cap context(synset)|$$

Word embeddings The second similarity measure compares contexts of entities $context(e)$ and of WordNet synsets $context(synset)$ (represented as vectors of words). The comparison is based on the distance of contexts in vector spaces. This method adopts the cosine distance between two words generated by the word embedding model to identify the similarity between them. We retrieve the similarity between $context(e)$ and $context(synset)$, then we calculate the average similarity. After calculating this average to all elements of the context, we calculate the average of the context, considering the context length. The synset with the higher average is selected.

4.2 Identification of correspondences to top-level ontologies

In this step we perform the identification of the top concept. This step relies on the representation of the given existing alignments.

DOLCE correspondence identification This step uses existing alignments between DOLCE-Lite-Plus and WordNet 1.6. For each concept of the domain ontology, we use the selected synset (step 1) to identify the corresponding concept in OntoWordNet. To select the concept in OntoWordNet we compare the WordNet synset with each concept c in OntoWordNet (recall that concepts are represented by the concatenation of words). A bag of words for the OntoWordNet concept is created from the concatenated words and gloss, i.e., $context(c)$. Then, we overlap the synset and c.

$$score'_{Lesk}(c) = |context(c) \cap context(synset)|$$

After finding the OntoWordNet concept c corresponding to the synset, the higher level concept h^c of c is retrieved,. h^c corresponds to the DOLCE concept (Figure 1).

SUMO correspondence identification Similarly to the correspondence identification in DOLCE, this step uses existing alignments between SUMO and WordNet 3.0, in order to identify the domain and top concepts correspondences. As SUMO-WordNet alignment is a file containing the synset ID, terms, gloss, and the alignment to top concept (§2.2), we search for the domain selected synset in this file and, if the synset is found, we match the domain concept with the top-level concept related to the synset.

As described above, our approach depends on the availability of alignments between the background knowledge resource (here, WordNet) and the top-level ontologies. Hence, we are able to exploit other top-level ontologies in case such alignments exist. This leads also to the question on the maintenance of these alignments with the evolution of the ontologies and the given resource, which is out of the scope of this work.

5 Experiments

5.1 Material and methods

Domain ontologies We consider a set of ontologies from three different domains. First, SSN (W3C Semantic Sensor Network Ontology) describes sensors, devices, observations, measurements and other terms, enabling reasoning of individual sensors and the connection of them. A recent version of SSN includes a lightweight core ontology called SOSA (Sensor, Observation, Sample, and Actuators). SSN is aligned to SOSA and both ontologies are aligned to DOLCE Ultralite (DUL). SSN is composed of 18 first level concepts, from those, 8 concepts are aligned to the top ontology DUL. CORA (Core Ontology for Robotic and Automation) [28] specifies the main concepts, relations, and axioms of robotics and automation domains. Second, CORA is aligned with SUMO top ontology. CORA ontology, considering all its modules (CoraX, Cora, RParts, and POS) is composed of 34 first level concepts, from which 29 of them are aligned to SUMO. Finally, seven ontologies from the OAEI Conference data set[5] have been used (Cmt, ConfTool, Edas, Ekaw, Iasted, Sigkdd, SofSem). These ontologies are involved in reference alignments. These ontologies sum up 501 concepts, however, we consider in our

---

[5] http://oaei.ontologymatching.org/2017/conference/index.html

experiments the first-level concept of the hierarchies, what corresponds to 70 concepts (assuming that the other concepts will inherit their alignment with top ontologies from their roots). The choice for these ontologies is motivated by the fact that they are either widely adopted in real world scenarios or in experiments regarding automatic ontology matching approaches.

WordNet top-level alignments We use DOLCE, SUMO, and existing WordNet to top-level ontology alignments (§2). These previous alignments have been developed by specialists, hence if the selected synset is correct, the top-level concept (aligned as super-concept of that synset) is assumed to be a super-concept of the domain concept.

Word embedding models We used pre-trained models, GloVe [27] and GoogleNews[6]. GloVe is an *unsupervised learning algorithm to obtain vector representations for words*[7]. The training phase uses the Wikipedia 2014 and Gigaword5 corpora. It has 6 billions tokens, 400 thousand vocabulary size and neural network dimension of 200. The GoogleNews model is trained on part of Google News dataset (about 100 billion words). The model contains 300-dimensional vectors for 3 million words and phrases.

OAEI 2017 tools Our baseline corresponds to the results of a set of matching tools participating in OAEI 2017, with exception only of those specialised in instance matching (Legato, I-match and njuLink) and one specialised in the bio domain (Yam-bio). The matchers that were tested in our experiment are: ALIN, AML, CroLOM, KEPLER, LogMap, LogMap-Lite, ONTMAT, POMap, SANOM, WikiV3, WikiMatch and XMap. The reader can refer to OAEI papers[8] for a detailed description of them. All tools were run with their default configuration settings. All generated correspondences are available in `https://github.com/danielasch/top-match`.

5.2 Results and discussion

We run our system with the Lesk similarity (*lesk*) and word embedding models (*WE-GloVe and WE-GlloogleNews*) and the OAEI tools for 16 matching tasks (SSN and DLP, CORA and SUMO, and 7 Conference ontologies with DLP and SUMO). All alignments generated by our approach are available online[9]. They have been evaluated in terms of precision and recall. With respect to the reference alignments, for the pairs involving SSN and CORA, given that these ontologies are already aligned to the top ontologies, we adopt these existing alignments as reference. We note that SSN is originally aligned with a different version of DOLCE. We hence consider the results in an interpreted way which consists at looking each generated correspondence and identify if they are the exact correspondence or related to the previous alignment via a subsumption relation. In the same way, we observe that some found correspondences from CORA and SUMO, were not exact the same of the adopted reference, however, they are hierarchically related, hence, we also adopted the interpreted evaluation.

For the Conference data set, which is not equipped with reference alignments to DOLCE and SUMO, the generated correspondences were manually evaluated by three

---

[6] https://code.google.com/archive/p/word2vec/

[7] https://nlp.stanford.edu/projects/glove/

[8] http://www.om2017.ontologymatching.org/#ap

[9] https://github.com/danielasch/top-match

specialists. Firstly, one evaluator analysed each correspondence, after, the results were discussed with all evaluators, maintaining or changing the initial analysis. For this data set, we made the hypothesis that, for each top domain concept, a corresponding WordNet synset exists. Hence, we are able to compute both precision and recall. As shown in Figure 2, the best results were obtained for the conference domain, with .80 of F-measure with WE-GloogleNews. We observe that overall WE-GloogleNews performs better than Lesk and WE-GloVe. However, looking at the SSN and CORA domain ontologies, the obtained results are lower than for Conference. Our hypothesis is that concepts from the conference ontology are more general (common sense) than these other domains. Note that the selected word embedding models were trained with general domain texts. The better performance obtained with the WE-GoogleNews model over the WE-GloVe model could be explained by the larger coverage of the first with respect to the training set.

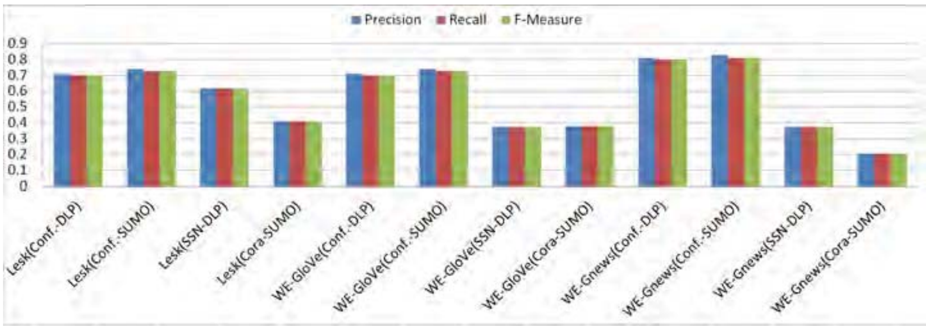


Fig. 2. Precision, recall and F-measure for each synset disambiguation strategy.

Regarding the number of correspondences, our approach was able to find 69 out of 70 correspondences from the Conference ontologies (we were not able to find the correspondences for 1 concept, for which there is no entry in WordNet) Considering Lesk and WE-GloVe, 51 correct correspondences were found when aligned to SUMO and 49 correct with DOLCE. This number increased up when using WE-GloogleNews (57 and 56, respectively). For SSN-DOLCE, we have 5 correct correspondences out of 8 considering Lesk, and 3 correct with WE-GloVe and WE-GloogleNews. For CORA-SUMO, 12 correct in a total of 29 correspondences considering Lesk, 11 correct with WE-GloVe and 6 correct with WE-GloogleNews.

Although our approach was able to found a high number of correspondences for the three domains, in some cases, the generated correspondences were wrong. First, as we adopt the context of concepts, this seems not to be enough to disambiguate the sense of the domain concept (Conference domain ontologies are not equipped of comments and labels). This can be improved by enriching the terminological layer. Second, we can observe that word embedding based on Google News model contributes to the disambiguation step, mainly with the Conference ontologies. However, for SSN and CORA it is still not able to retrieve the right synsets. In order to overcome this weaknesses, one direction is to use domain-specific embedding models. Third, the word sense disambiguation here is still based on the overlapping of words, and word sense disambiguation techniques could be used instead.

OAEI 2017 matching tools Only 4 tools (AML, LogMap, LogMapLite, and POMap) were able to find correspondences for 6 pairs of ontologies. Considering the correspondences found by these tools, 13 domain concepts from conference (out of 70) were aligned. Regarding the number of correspondences, *AML* was able to find 12 correspondences, and 7 of them were correct. *POMap* found 7 correspondences, and 6 were correct. *LogMap* and *LogMapLite* found 6 correspondences respectively, and 5 of them were correct. Figure 3 presents precision, recall and F-measure for each tool (including our evaluated techniques). Related to CORA, 1 correspondence was correctly found by *POMap*. As shown, our approach outperforms all system in terms of Recall and F-measure. Looking at *WE-GloogleNews*, the results are quite similar in terms of precision and better than all in terms of recall and F-measure. As somehow expected, while the tools perform well in terms of precision, they retrieve a limited number of correspondences.

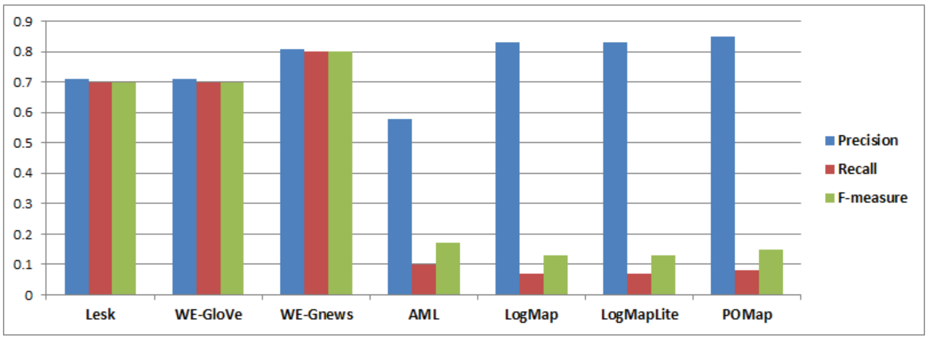


Fig. 3. Precision, recall and F-measure from each matching tool.

6 Concluding remarks and future work

This paper presented an approach to match domain and top-level ontologies, exploiting alignments between WordNet and top ontologies. Our evaluation was based on ontologies from three domains with DOLCE and SUMO top-level ontologies. Overall, we consider that existing top-level and WordNet alignments is a valuable resource for the task, at least for certain general domains. For most of the concepts from the domain ontologies we found a correspondence with the top ontology. We have evaluated OAEI matching tools in this task and, as expected, our approach outperforms all of them. Even though they were not exactly developed for that purpose, their results were the only available for comparison, and we set that as a baseline. To the best of our knowledge, our approach is the first attempt towards automatizing the process of aligning top and domain ontologies. As future work, we plan to provide a reference alignment involving the OAEI Conference dataset and DOLCE and SUMO ontologies with the aim of proposing a OAEI track for this task involving top and domain ontologies. We plan as well to combine Wordnet measures with other distributional semantics approaches and adopt other background knowledge resources as BabelNet.

This work will be published as part of the book "Emerging Topics in Semantic Technologies". ISWC 2018 Satellite Events. E. Demidova, A.J. Zaveri, E. Simperl (Eds.), ISBN: 978-3-89838-736-1, 2018, AKA Verlag Berlin.

References

1. Brodaric, B., Probst, F.: DOLCE ROCKS: Integrating Geoscience Ontologies with DOLCE. In: Semantic Scientific Knowledge Integration. pp. 3–8 (2008)
2. David, J.: AROMA results for OAEI 2011. In: Proc. of the 6th Intern. Workshop on Ontology Matching. pp. 122–125 (2011)
3. Euzenat, J., Shvaiko, P.: Ontology Matching. Springer-Verlag, 1st edn. (2007)
4. Gangemi, A., Guarino, N., Masolo, C., Oltramari, A.: Sweetening WORDNET with DOLCE. AI Magazine 24(3), 13–24 (2003)
5. Gangemi, A., Guarino, N., Masolo, C., Oltramari, A., Schneider, L.: Sweetening Ontologies with DOLCE. In: Proc. of the 13th Intern. Conf. on Knowledge Engineering and Knowledge Management. pp. 166–181 (2002)
6. Gangemi, A., Navigli, R., Velardi, P.: The OntoWordNet Project: Extension and Axiomatization of Conceptual Relations in WordNet, pp. 820–838. Springer Berlin Heidelberg (2003)
7. Guarino, N.: Formal Ontology in Information Systems: Proc. of the 1st Intern. Conference. IOS Press, 1st edn. (1998)
8. Jung, J.J., Håkansson, A., Hartung, R.: Indirect Alignment between Multilingual Ontologies: A Case Study of Korean and Swedish Ontologies. In: Agent and Multi-Agent Systems: Technologies and Applications: 3rd KES Intern. Symposium. pp. 233–241 (2009)
9. Kachroudi, M., Chelbi, A., Souid, H., Ben Yahia, S.: Fiona: A framework for indirect ontology alignment. In: Foundations of Intelligent Systems. pp. 224–229 (2015)
10. Khan, Z., Keet, C.M.: The Foundational Ontology Library ROMULUS. In: Proc. of the 3rd Conf. on Model and Data Engineering. pp. 200–211 (2013)
11. Khan, Z., Keet, C.M.: Toward semantic interoperability with aligned foundational ontologies in ROMULUS. In: Proc. of the 7th Conf. on Knowledge Capture. pp. 23–26 (2013)
12. Kwak, J., Yong, H.: Ontology Matching based on Hypernym, Hyponym, Holonym and Meronym Sets in WordNet. Web and Semantic Technology 1(2), 1–14 (2010)
13. Lesk, M.: Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone. In: Proc. of the 5th Intern. Conf. on Systems Documentation. pp. 24–26 (1986)
14. Lin, F., Sandkuhl, K.: A Survey of Exploiting WordNet in Ontology Matching. In: Proc. of the 20th Intern. Federation for Information Processing, pp. 341–350 (2008)
15. Mascardi, V., Cordì, V., Rosso, P.: A Comparison of Upper Ontologies. In: Proc. of the 8th AI*IA/TABOO Joint Workshop on Agents and Industry. pp. 55–64 (2007)
16. Mascardi, V., Locoro, A., Rosso, P.: Automatic Ontology Matching via Upper Ontologies: A Systematic Evaluation. IEEE Trans. on Knowl. and Data Eng. 22(5), 609–623 (2010)
17. Masolo, C., Borgo, S., Gangemi, A., Guarino, N., Oltramari, A.: WonderWeb deliverable D18 ontology library (final). Tech. rep., IST Project 2001-33052 WonderWeb: Ontology Infrastructure for the Semantic Web (2003)
18. Mika, P., Oberle, D., Gangemi, A., Sabou, M.: Foundations for Service Ontologies: Aligning OWL-S to Dolce. In: Proc. of the 13th Conf. on World Wide Web. pp. 563–572 (2004)

19. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. CoRR (2013)
20. Miller, G.A.: WordNet: A Lexical Database for English. Commun. ACM 38(11), 39–41 (1995)
21. Navigli, R.: Word Sense Disambiguation: A Survey. ACM Comput. Surv. 41(2), 10:1–10:69 (2009)
22. Niles, I., Pease, A.: Towards a Standard Upper Ontology. In: Proc. of the Intern. Conf. on Formal Ontology in Information Systems. pp. 2–9 (2001)
23. Niles, I., Pease, A.: Linking Lexicons and Ontologies: Mapping WordNet to the Suggested Upper Merged Ontology. In: Proc. of the Intern. Conf. on Information and Knowledge Engineering. pp. 412–416 (2003)
24. Oberle, D., Ankolekar, A., Hitzler, P., et al.: DOLCE ergo SUMO: On foundational and domain models in the SmartWeb Integrated Ontology (SWIntO). Web Semantics: Science, Services and Agents on the World Wide Web 5(3) (2007)
25. Padilha, N.F., Baião, F., Revoredo, K.: Alignment Patterns based on Unified Foundational Ontology. In: Proc. of the Brazilian Ontology Research Seminar. pp. 48–59 (2012)
26. Paulheim, H., Gangemi, A.: Serving DBpedia with DOLCE - More than Just Adding a Cherry on Top. In: The Semantic Web, pp. 180–196 (2015)
27. Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word representation. In: Empirical Methods in Natural Language Processing. pp. 1532–1543 (2014)
28. Prestes, E., Carbonera, J.L., Fiorini, S.R., et al.: Towards a core ontology for robotics and automation. Robotics and Autonomous Systems 61(11), 1193 – 1204 (2013)
29. Reed, S., Lenat, D.: Mapping Ontologies into Cyc. In: Proc. of the Workshop on Ontologies For The Semantic Web. pp. 1–6 (2002)
30. Schadd, F.C., Roos, N.: Coupling of Wordnet Entries for Ontology Mapping Using Virtual Documents. In: Proc. of the 7th Conf. on Ontology Matching. pp. 25–36 (2012)
31. Schmidt, D., Basso, R., Trojahn, C., Vieira, R.: Matching Domain and Top-level Ontologies via OntoWordNet. In: Workshop on Ontology Matching. pp. 1–2 (2017)
32. Schmidt, D., Trojahn, C., Vieira, R.: Analysing Top-level and Domain Ontology Alignments from Matching Systems. In: Workshop on Ontology Matching. pp. 1–12 (2016)
33. Semy, S., Pulvermacher, M., Obrst, L.: Toward the use of an upper ontology for U.S. government and U.S. military domains: An evaluation. Tech. rep., Sub. to Workshop on Information Integration on the Web, in conjunction with VLDB-2004 (2004)
34. Seppala, S.: Mapping WordNet to Basic Formal Ontology using the KYOTO ontology. In: Proc. of the Intern. Conf. on Biomedical Ontology. pp. 1–2 (2015)
35. Silva, V., Campos, M., Silva, J., Cavalcanti, M.: An Approach for the Alignment of Biomedical Ontologies based on Foundational Ontologies. Information and Data Management 2(3), 557–572 (2011)
36. Vieira, R., Revoredo, K.: Using word semantics on entity names for correspondence set generation. In: Workshop on Ontology Matching. pp. 223–224 (2017)
37. Wang, P.: Lily Results on SEALS Platform for OAEI 2011. In: Proc. of the 6th Intern. Workshop on Ontology Matching. pp. 156–162 (2011)
38. Yatskevich, M., Giunchiglia, F.: Element level semantic matching using WordNet. In: Meaning Coordination and Negotiation Workshop. pp. 37–48 (2004)
39. Zhang, S., Bodenreider, O.: Alignment of multiple ontologies of anatomy: Deriving indirect mappings from direct mappings to a reference. In: American Medical Informatics Association Annual Symposium (2005)
40. Zhang, Y., Wang, X., Lai, S., He, S., Liu, K., Zhao, J., Lv, X.: Ontology matching with word embeddings. In: Sun, M., Liu, Y., Zhao, J. (eds.) Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data. pp. 34–45 (2014)

We Divide, You Conquer:

From Large-scale Ontology Alignment to Manageable Subtasks with a Lexical Index and Neural Embeddings[⋆]

Ernesto Jiménez-Ruiz[1,2], Asan Agibetov[3], Matthias Samwald[3], Valerie Cross[4]

[1] The Alan Turing Institute, London, United Kingdom
[2] Department of Informatics, University of Oslo, Norway
[3] Section for Artificial Intelligence and Decision Support, Medical University of Vienna, Vienna, Austria
[4] Miami University, Oxford, OH 45056, United States

Abstract. Large ontologies still pose serious challenges to state-of-the-art ontology alignment systems. In this paper we present an approach that combines a lexical index, a neural embedding model and locality modules to effectively divide an input ontology matching task into smaller and more tractable matching subtasks. We have conducted a comprehensive evaluation using the datasets of the Ontology Alignment Evaluation Initiative. The results are encouraging and suggest that the proposed methods are adequate in practice and can be integrated within the workflow of state-of-the-art systems.

1 Introduction

Large-scale ontology matching tasks still pose serious challenges to ontology alignment systems. For example, only 6 out of 10 systems participating in the OAEI 2017 *largebio track* were able to complete the largest tasks [2]. OAEI systems are typically able to cope with small and medium size ontologies, but fail to complete large tasks in a given time frame and/or with the available resources (*e.g.*, memory). Prominent examples across the OAEI campaigns are: *(i)* YAM++ version 2011 [3] (best results in *conference* track, but failed to complete the *anatomy* task); *(ii)* CODI version 2011.5 [4] (best results in *anatomy* but could not cope with the *largebio* track); *(iii)* MAMBA version 2015 [5] (top system in the *conference* track but could not complete the *anatomy* track); *(iv)* FCA-Map version 2016 [6] (completed both *anatomy* and *phenotype* tasks but did not complete the largest *largebio* tasks); and *(v)* POMap version 2017 [7] (one of the top systems in *anatomy* but could not finish the largest *largebio* tasks).

In this paper we propose a novel method to effectively divide the matching task into several (independent) smaller subtasks. This method relies on an efficient lexical index (as in LogMap [8]), a neural embedding model [9] and locality modules [10]. Unlike other state-of-the-art approaches, our method provides guarantees about the preservation of the *coverage* of the relevant ontology alignments as defined in Section 2.2.

2 Preliminaries

In this section we introduce the background concepts that are used throughout the paper.

[⋆] An extended version of this paper is available in arXiv.org [1].

2.1 Basic definitions

A *mapping* (also called *match* or *correspondence*) between entities[1] of two ontologies[2] $\mathcal{O}_1$ and $\mathcal{O}_2$ is typically represented as a 4-tuple $\langle e_1, e_2, r, c\rangle$ where e_1 and e_2 are entities of $\mathcal{O}_1$ and $\mathcal{O}_2$, respectively; $r \in \{\sqsubseteq, \sqsupseteq, \equiv\}$ is a semantic relation; and c is a confidence value, usually, a real number within the interval $(0, 1]$. In our approach we simply consider mappings as a pair $\langle e_1, e_2\rangle$. An ontology *alignment* is a set of mappings $\mathcal{M}$ between two ontologies $\mathcal{O}_1$ and $\mathcal{O}_2$.

An ontology *matching task* $\mathcal{MT}$ is composed of a pair of ontologies $\mathcal{O}_1$ (typically called source) and $\mathcal{O}_2$ (typically called target) and possibly an associated *reference alignment* $\mathcal{M}^{RA}$. The objective of a matching task is to discover an (implicit) overlapping of $\mathcal{O}_1$ and $\mathcal{O}_2$ in the form of an alignment $\mathcal{M}$. The *size or search space* of a matching task is typically bound to the size of the Cartesian product between the entities of the input ontologies: $|Sig(\mathcal{O}_1)| \times |Sig(\mathcal{O}_2)|$ being $Sig(\mathcal{O})$ the signature (*i.e.*, entities) of the ontology $\mathcal{O}$.

An ontology *matching system* is a program that, given as input the ontologies $\mathcal{O}_1$ and $\mathcal{O}_2$ of a matching task, generates an ontology alignment $\mathcal{M}^S$.

The standard evaluation measures for an alignment $\mathcal{M}^S$ are *precision* (P), *recall* (R) and *f-measure* (F) computed against a reference alignment $\mathcal{M}^{RA}$ as follows:

$$P = \frac{|\mathcal{M}^S \cap \mathcal{M}^{RA}|}{|\mathcal{M}^S|},\ R = \frac{|\mathcal{M}^S \cap \mathcal{M}^{RA}|}{|\mathcal{M}^{RA}|},\ F = 2 \cdot \frac{P \cdot R}{P + R} \tag{1}$$

2.2 Matching subtasks and quality measures: size ratio and coverage

We denote *division* of an ontology matching task $\mathcal{MT}$, composed by the ontologies $\mathcal{O}_1$ and $\mathcal{O}_2$, as the process of finding matching subtasks $\mathcal{MT}_i = \langle \mathcal{O}_1^i, \mathcal{O}_2^i\rangle$ (with i=1,…,n), where $\mathcal{O}_1^i \subset \mathcal{O}_1$ and $\mathcal{O}_2^i \subset \mathcal{O}_2$. The size of the matching subtasks aims at being smaller than the original task in terms of search space. Let $\mathcal{D}^n_{\mathcal{MT}} = \{\mathcal{MT}_1, \ldots, \mathcal{MT}_n\}$ be the result of dividing a matching task $\mathcal{MT}$. The *size ratios* of the matching subtasks $\mathcal{MT}_i$ and $\mathcal{D}^n_{\mathcal{MT}}$ are computed as follows:

$$\mathsf{SizeRatio}(\mathcal{MT}_i, \mathcal{MT}) = \frac{|Sig(\mathcal{O}_1^i)| \times |Sig(\mathcal{O}_2^i)|}{|Sig(\mathcal{O}_1)| \times |Sig(\mathcal{O}_2)|} \tag{2}$$

$$\mathsf{SizeRatio}(\mathcal{D}^n_{\mathcal{MT}}, \mathcal{MT}) = \sum_{i=1}^{n} \mathsf{SizeRatio}(\mathcal{MT}_i, \mathcal{MT}) \tag{3}$$

The ratio $\mathsf{SizeRatio}(\mathcal{MT}_i, \mathcal{MT})$ is expected to be less than 1.0 while the aggregation $\sum_{i=1}^{n} \mathsf{SizeRatio}(\mathcal{MT}_i, \mathcal{MT})$, being n the number of matching subtasks, can be greater than 1.0 (as matching subtasks may overlap).

The *coverage* of the matching subtask aims at providing guarantees about the preservation of the (potential) outcomes of the original matching task (*i.e.*, information loss). That is, it indicates if the relevant ontology alignments in the original matching task can still be computed with the matching subtasks. The coverage is calculated with respect to a relevant alignment $\mathcal{M}$, possibly the reference alignment $\mathcal{M}^{RA}$ of the matching task if it exists. The formal notion of coverage is given in Definitions 1 and 2.

---

[1] We refer to (OWL 2) classes, data and object properties and named individuals as entities.

[2] We assume ontologies are expressed in OWL 2.

Definition 1 (Coverage of a matching task). *Let $\mathcal{MT} = \langle \mathcal{O}_1, \mathcal{O}_2 \rangle$ be a matching task and $\mathcal{M}$ an alignment. We say that a mapping $m = \langle e_1, e_2 \rangle \in \mathcal{M}$ is covered by the matching task if $e_1 \in Sig(\mathcal{O}_1)$ and $e_2 \in Sig(\mathcal{O}_2)$. The coverage of $\mathcal{MT}$ w.r.t. $\mathcal{M}$ (denoted as $\mathsf{Coverage}(\mathcal{MT}, \mathcal{M})$) represents the set of mappings $\mathcal{M}' \subseteq \mathcal{M}$ covered by $\mathcal{MT}$.*

Definition 2 (Coverage of the matching task division). *Let the result of dividing a matching task $\mathcal{MT}$ be $\mathcal{D}^n_{\mathcal{MT}} = \{\mathcal{MT}_1, \ldots, \mathcal{MT}_n\}$ and $\mathcal{M}$ an alignment. We say that a mapping $m \in \mathcal{M}$ is covered by $\mathcal{D}_{\mathcal{MT}}$ if m is at least covered by one of the matching subtask $\mathcal{MT}_i$ (with i=1,...,n) as in Definition 1. The coverage of $\mathcal{D}_{\mathcal{MT}}$ w.r.t. $\mathcal{M}$ (denoted as $\mathsf{Coverage}(\mathcal{D}_{\mathcal{MT}}, \mathcal{M})$) represents the set of mappings $\mathcal{M}' \subseteq \mathcal{M}$ covered by $\mathcal{D}_{\mathcal{MT}}$. The coverage is often given as a ratio with respect to the (covered) alignment:*

$$\mathsf{CoverageRatio}(\mathcal{D}^n_{\mathcal{MT}}, \mathcal{M}) = \frac{|\mathsf{Coverage}(\mathcal{D}_{\mathcal{MT}}, \mathcal{M})|}{|\mathcal{M}|} \tag{4}$$

2.3 Locality-based modules in ontology alignment

Logic-based module extraction techniques compute ontology fragments that capture the meaning of an input signature with respect to a given ontology. In this paper we rely on bottom-locality modules [10], which will be referred to as locality-modules or simply as modules. Locality modules play an important role in ontology alignment tasks. For example, they provide the context, *i.e.*, sets of *semantically related* entities [10], for the entities in a given mapping or set of mappings as formally presented in Definition 3.

Definition 3 (Context of a mapping and an alignment). *Let $m = \langle e_1, e_2 \rangle$ be a mapping between two ontologies $\mathcal{O}_1$ and $\mathcal{O}_2$. We define the context of m (denoted as $\mathsf{Context}(m, \mathcal{O}_1, \mathcal{O}_2)$) as a pair of modules $\mathcal{O}'_1 \subseteq \mathcal{O}_1$ and $\mathcal{O}'_2 \subseteq \mathcal{O}_2$, where $\mathcal{O}'_1$ and $\mathcal{O}'_2$ include the semantically related entities to e_1 and e_2, respectively [10]. Similarly, the* context *for an alignment $\mathcal{M}$ between two ontologies $\mathcal{O}_1$ and $\mathcal{O}_2$ is denoted as $\mathsf{Context}(\mathcal{M}, \mathcal{O}_1, \mathcal{O}_2) = \langle \mathcal{O}'_1, \mathcal{O}'_2 \rangle$, where $\mathcal{O}'_1$ and $\mathcal{O}'_2$ are modules including the semantically related entities for the entities $e_1 \in Sig(\mathcal{O}_1)$ and $e_2 \in Sig(\mathcal{O}_2)$ in each mapping $m = \langle e_1, e_2 \rangle \in \mathcal{M}$.*

2.4 Context as matching task

The context of an alignment between two ontologies represents the (explicit) overlapping of these ontologies with respect to the aforesaid alignment. Intuitively, the ontologies in the context of an alignment cover all the mappings in that alignment. Definition 4 formally presents the context of an alignment as the *overlapping* matching task to discover that alignment.

Definition 4 (Overlapping matching task). *Let $\mathcal{M}$ be an alignment between $\mathcal{O}_1$ and $\mathcal{O}_2$, and $\mathsf{Context}(\mathcal{M}, \mathcal{O}_1, \mathcal{O}_2) = \langle \mathcal{O}'_1, \mathcal{O}'_2 \rangle$ the context of $\mathcal{M}$. We define $\mathcal{MT}^{\mathcal{M}}_{\mathcal{O}_1\text{-}\mathcal{O}_2} = \langle \mathcal{O}'_1, \mathcal{O}'_2 \rangle$ as the overlapping matching task for $\mathcal{M}$. A matching task $\mathcal{MT} = \langle \mathcal{O}_1, \mathcal{O}_2 \rangle$ can be reduced to the task $\mathcal{MT}^{\mathcal{M}}_{\mathcal{O}_1\text{-}\mathcal{O}_2} = \langle \mathcal{O}'_1, \mathcal{O}'_2 \rangle$ without information loss in terms of finding $\mathcal{M}$.*

A matching system should aim at computing $\mathcal{M}$ with both the original matching task $\mathcal{MT}$ and the reduced task $\mathcal{MT}^{\mathcal{M}}_{\mathcal{O}_1\text{-}\mathcal{O}_2}$. For example, in the small OAEI *largebio* tasks [2] systems are given, instead of the original matching task (*e.g.*, whole FMA and NCI ontologies), the context of the reference alignment as a (reduced) overlapping matching task (*e.g.*, $\mathcal{MT}^{RA}_{\text{fma-nci}} = \mathsf{Context}(\mathcal{M}^{RA}_{\text{fma-nci}}, \mathcal{O}_{\text{FMA}}, \mathcal{O}_{\text{NCI}}) = \langle \mathcal{O}'_{\text{FMA}}, \mathcal{O}'_{\text{NCI}} \rangle$).

Table 1: Inverted lexical index LexI (left) and entity index (right). For readability, stemming techniques have not been applied and index values have been split into elements of $\mathcal{O}_1$ and $\mathcal{O}_2$. '-' indicates that the ontology does not contain entities for that entry.

Index key	**Index value**	
	Entities $\mathcal{O}_1$	**Entities $\mathcal{O}_2$**
{ acinus }	7661,8171	118081
{ mesothelial, pleural }	19987	117237
{ hamate, lunate }	55518	-
{ feed, breast }	-	113578,111023

ID	**URI**
7661	$\mathcal{O}_1$:Serous_acinus
8171	$\mathcal{O}_1$:Hepatic_acinus
19987	$\mathcal{O}_1$:Mesothelial_cell_of_pleura
55518	$\mathcal{O}_1$:Lunate_facet_of_hamate
118081	$\mathcal{O}_2$:Liver_acinus
117237	$\mathcal{O}_2$:Pleural_Mesothelial_Cell
113578	$\mathcal{O}_2$:Breast_Feeding
111023	$\mathcal{O}_2$:Inability_To_Breast_Feed

3 Methods

The approach presented in this paper relies on an 'inverted' lexical index (we will refer to this index as LexI), commonly used in information retrieval applications, and also used in ontology alignment systems like LogMap [8].

3.1 The lexical index LexI

LexI encodes the labels of all entities of the input ontologies $\mathcal{O}_1$ and $\mathcal{O}_2$, including their lexical variations (*e.g.*, preferred labels, synonyms), in the form of pairs *key-value* where the key is a set of words and the value is a set of entity identifiers[3] such that the set of words of the key appears in (one of) the entity labels. Table 1 shows a few example entries of LexI for two input ontologies.

LexI is created as follows. *(i)* Each label associated to an ontology entity is split into a set of words; for example, the label "Lunate facet of hamate" is split into the set {"lunate", "facet", "of", "hamate"}. *(ii)* Stop-words are removed, for example,"of" is removed from the set of words (*i.e.*, {"lunate", "facet", "hamate"}). *(iii)* Stemming techniques are applied to each word (*i.e.*, {"lunat", "facet", "hamat"}). *(iv)* Combinations of (sub)set of words serve as keys in LexI; for example, {"lunat", "facet"}, {"hamat", "lunat"} and so on.[4] *(v)* Entities leading to the same (sub)set of words are associated to the same key in LexI, for example, the entity $\mathcal{O}_1$:Lunate_facet_of_hamate with numerical identifier 55518 is associated to the LexI key {"hamat", "lunat"} (see Table 1). Finally, *(vi)* entries in LexI pointing to entities of only one ontology are not considered (see last two rows of LexI in Table 1). Note that a single entity label may lead to several entries in LexI, and each entry in LexI points to one or many entities.

Each entry in LexI, after discarding entries pointing to only one ontology, is a source of candidate mappings. For instance the example in Table 1 suggests that there is a (potential) mapping $m = \langle \mathcal{O}_1{:}\mathsf{Serous\_acinus}, \mathcal{O}_2{:}\mathsf{Liver\_acinus}, \equiv, c \rangle$ since the entities $\mathcal{O}_1$:Serous_acinus and $\mathcal{O}_2$:Liver_acinus are associated to the same entry in LexI $\{$*acinus*$\}$. These mappings are not necessarily correct but link lexically-related entities, that is, those entities sharing at least one word among their labels (*e.g.*, "acinus"). Given a subset of entries of LexI (*i.e.*, $l \subseteq \mathsf{LexI}$), the function $\mathsf{Mappings}(l) = \mathcal{M}^l$ provides the set of mappings derived from l. We refer to the set of all (potential) mappings suggested by LexI (*i.e.*, Mappings(LexI)) as $\mathcal{M}^{\mathsf{LexI}}$. Note that $\mathcal{M}^{\mathsf{LexI}}$ represents a manageable subset of the Cartesian product between the entities of the input ontologies.

---

[3] The indexation module associates unique numerical identifiers to entity URIs.

[4] In order to avoid a combinatorial blow-up, the number of computed subsets of words is limited.

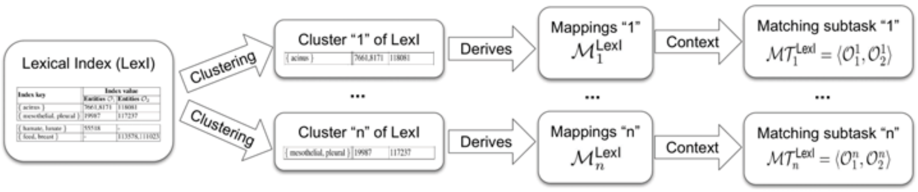


Fig. 1: Pipeline to extract matching subtasks from LexI.

Most of the state-of-the-art ontology matching systems rely, in one way or another, on lexical similarity measures to either discover or validate candidate mappings [11]. Thus, mappings outside $\mathcal{M}^{\mathsf{LexI}}$ will rarely be discovered by standard matching systems.

3.2 Creation of matching subtasks from LexI

Considering all entries in LexI (*i.e.*, one cluster) may lead to a very large number of candidate mappings $\mathcal{M}^{\mathsf{LexI}}$. The context of $\mathcal{M}^{\mathsf{LexI}}$ leads to (two) large overlapping modules $\mathcal{O}_1^{\mathsf{LexI}}$ and $\mathcal{O}_2^{\mathsf{LexI}}$ that, although smaller than the input ontologies $\mathcal{O}_1$ and $\mathcal{O}_2$, may still be challenging for many ontology matching systems. A solution is to divide the entries in LexI in more than one cluster.

Definition 5 (Matching subtasks from LexI). *Let $\mathcal{MT} = \langle \mathcal{O}_1, \mathcal{O}_2 \rangle$ be a matching task, LexI the lexical index of the ontologies $\mathcal{O}_1$ and $\mathcal{O}_2$, and $\{c_1, \ldots, c_n\}$ n clusters of entries in LexI. We denote the set of matching subtasks from LexI as $\mathcal{D}^n_{\mathcal{MT}} = \{\mathcal{MT}_1^{\mathsf{LexI}}, \ldots, \mathcal{MT}_n^{\mathsf{LexI}}\}$ where each cluster c_i leads to the matching subtask $\mathcal{MT}_i^{\mathsf{LexI}} = \langle \mathcal{O}_1^i, \mathcal{O}_2^i \rangle$, such that* $\mathsf{Mappings}(c_i) = \mathcal{M}_i^{\mathsf{LexI}}$ *is the set of mappings suggested by the LexI entries in c_i and $\mathcal{O}_1^i$ and $\mathcal{O}_2^i$ represent the context of $\mathcal{M}_i^{\mathsf{LexI}}$ w.r.t. $\mathcal{O}_1$ and $\mathcal{O}_2$.*

Figure 1 shows an overview of the pipeline where LexI is split into n clusters and these clusters lead to n matching subtasks $\mathcal{D}^n_{\mathcal{MT}} = \{\mathcal{MT}_1^{\mathsf{LexI}}, \ldots, \mathcal{MT}_n^{\mathsf{LexI}}\}$.[5]

Hypothesis 1 *If $\mathcal{MT} = \langle \mathcal{O}_1, \mathcal{O}_2 \rangle$ is a matching task and $\mathcal{M}^S$ the mappings computed for $\mathcal{MT}$ by a lexical-based matching system, then, with independence of the clustering strategy of LexI and the number of subtasks n, $\mathcal{D}^n_{\mathcal{MT}} = \{\mathcal{MT}_1^{\mathsf{LexI}}, \ldots, \mathcal{MT}_n^{\mathsf{LexI}}\}$ will cover (almost) all the mappings in $\mathcal{M}^S$ (i.e.,* $\mathsf{CoverageRatio}(\mathcal{D}^n_{\mathcal{MT}}, \mathcal{M}^S) \approx 1.0$*).*

Hypothesis 1 suggests that a matching system will unlikely discover mappings with $\mathcal{MT} = \langle \mathcal{O}_1, \mathcal{O}_2 \rangle$ that cannot be discovered with $\mathcal{D}^n_{\mathcal{MT}} = \{\mathcal{MT}_1^{\mathsf{LexI}}, \ldots, \mathcal{MT}_n^{\mathsf{LexI}}\}$. This intuition is supported not only by the observation that most of the ontology matching systems rely on lexical similarity, but also by the use of the notion of context (see Definition 3 and Definition 4) in the creation of the matching subtasks.

Intuitively each cluster of LexI leads to a smaller set of mappings $\mathcal{M}_i^{\mathsf{LexI}}$ (with respect to $\mathcal{M}^{\mathsf{LexI}}$) and to a smaller matching task $\mathcal{MT}_i^{\mathsf{LexI}}$ (with respect to both $\mathcal{MT}^{\mathsf{LexI}}$ and $\mathcal{MT}$) in terms of search space. Hence $\mathsf{SizeRatio}(\mathcal{MT}_i^{\mathsf{LexI}}, \mathcal{MT})$ is expected to be smaller than 1.0, as mentioned in Section 2.2. Reducing the search space in each matching subtask $\mathcal{MT}_i^{\mathsf{LexI}}$ has the potential of enabling the use of systems that can not cope with the original matching task $\mathcal{MT}$ in a given time-frame or with (limited) computational resources. The aggregation of ratios may be greater than 1.0 and will depend on the clustering strategy.

---

[5] The number of clusters n is a parameter given as input. See Section 6 for a discussion of possibles ways of automatically obtaining n.

Hypothesis 2 *Given a matching task* $\mathcal{MT}$ *and an ontology matching system that fails to complete* $\mathcal{MT}$ *under a set of given computational constraints, there exists a division of the matching task* $\mathcal{D}^n_{\mathcal{MT}} = \{\mathcal{MT}_1^{\mathsf{LexI}}, \ldots, \mathcal{MT}_n^{\mathsf{LexI}}\}$ *for which that system is able to compute an alignment of the individual matching subtasks* $\mathcal{MT}_1^{\mathsf{LexI}}, \ldots, \mathcal{MT}_n^{\mathsf{LexI}}$ *under the same constraints.*

3.3 Clustering strategies

We have implemented two clustering strategies which we refer to as: *naive* and *neural embedding*. Both strategies receive as input the index LexI and the number of desired clusters n, and provide as output a set of clusters $\{c_1, \ldots, c_n\}$ from LexI. As in Definition 5, these clusters lead to the matching subtasks in $\mathcal{D}^n_{\mathcal{MT}} = \{\mathcal{MT}_1^{\mathsf{LexI}}, \ldots, \mathcal{MT}_n^{\mathsf{LexI}}\}$.

The choice of strategy, according to Hypothesis 1, will not have an impact on the coverage; but it may influence the size of the matching subtasks. Note that, neither of the strategies aims at computing optimal clusters of the entries in LexI, but clusters that can be efficiently computed.

Naive strategy. This strategy implements a very simple algorithm that randomly splits the entries in LexI into a given number of clusters of the same size. The matching tasks resulting from this strategy are expected to have a high overlapping as different entries in LexI leading to similar set of mappings may fall into different clusters. Although the overlapping of matching subtasks will impact the global search space, it is still expected to be smaller than in the original matching task.

Neural embedding strategy. This strategy aims at identifying more accurate clusters, leading to matching tasks with less overlapping, and thus, reducing the global size of the computed division of the matching task $\mathcal{D}^n_{\mathcal{MT}}$. It relies on StarSpace toolkit[6] and its neural embedding model [9], which aims at learning *entity embeddings*. Each entity[7] is described by a finite set of discrete *features* (bag-of-features). The model is trained by assigning a d-dimensional vector to each of the discrete features in the set that we want to embed directly. Applied to the lexical index LexI, the neural embedding model would learn vector representations for the individual words in the index keys, and for the individual entity identifiers in the index values. Since an index key is a set of words (see Table 1), we use the *mean vector* representation of the vectors associated to each word. Based on these *aggregated* neural embeddings we then perform standard clustering with the K-means algorithm.

Hypothesis 3 *There exists a number of clusters or matching subtasks 'n' for which the clustering strategies can compute* $\mathcal{D}^n_{\mathcal{MT}} = \{\mathcal{MT}_1^{\mathsf{LexI}}, \ldots, \mathcal{MT}_n^{\mathsf{LexI}}\}$ *for a given matching task* $\mathcal{MT}$ *such that* $\mathsf{SizeRatio}(\mathcal{D}^n_{\mathcal{MT}}, \mathcal{MT}) < 1.0$.

Hypothesis 3 suggests that there exists a division $\mathcal{D}^n_{\mathcal{MT}}$ of $\mathcal{MT}$ such that the size (or search space) of $\mathcal{D}^n_{\mathcal{MT}}$ is smaller than $\mathcal{MT}$, and $\mathcal{D}^n_{\mathcal{MT}}$ can be computed by the proposed naive and neural embedding strategies.

Table 2: OAEI matching tasks. Phenotype ontologies downloaded from BioPortal.

OAEI track	Source of $\mathcal{M}^{RA}$	Task	Ontology	Version	Size (classes)
Anatomy	Manually created	AMA-NCIA	AMA	v.2007	2,744
			NCIA	v.2007	3,304
Largebio	UMLS-Metathesaurus	FMA-NCI	FMA	v.2.0	78,989
		FMA-SNOMED	NCI	v.08.05d	66,724
		SNOMED-NCI	SNOMED	v.2009	306,591
Phenotype	Consensus alignment (vote=2) [12]	HPO-MP	HPO	v.2016-BP	11,786
			MP	v.2016-BP	11,721
		DOID-ORDO	DOID	v.2016-BP	9,248
			ORDO	v.2016-BP	12,936

4 Evaluation

In this section we support Hypothesis 1-3 (Section 3). We rely on the datasets of the Ontology Alignment Evaluation Initiative (OAEI) [2], more specifically, on the matching tasks provided in the *anatomy*, *largebio* and *phenotype* tracks (see Table 2).

The methods have been implemented in Java[8] and Python[9] and were tested on a Ubuntu Laptop with an Intel Core i7-4600U CPU@2.10GHz (4 cores). Up to 15 Gb of RAM was allocated. The next sections present the performed experiments.[10]

4.1 Adequacy of the clustering strategies

We have evaluated the adequacy of the clustering strategies to compute divisions $\mathcal{D}^n_{\mathcal{MT}} = \{\mathcal{MT}_1^{\mathsf{LexI}}, \ldots, \mathcal{MT}_n^{\mathsf{LexI}}\}$ for each of the matching tasks in Table 2 with respect to the available reference alignments. We report results in terms of coverage (as in Equation 4) and size (as in Equation 3) of the resulting division $\mathcal{D}^n_{\mathcal{MT}}$ of the matching tasks.

We have compared the two strategies for different number of clusters or resulting matching subtasks $n \in \{2, 5, 10, 20, 50, 100, 200\}$. For the naive strategy, as a random split of LexI is performed, we run 10 experiments for each of the values of n to evaluate the effect of different random selections. The variations in the size of the obtained matching tasks was negligible. Results represent the average of the 10 experiments

Coverage ratio. Figure 2 shows the coverage of the different divisions $\mathcal{D}^n_{\mathcal{MT}}$ of the matching task for the naive (left) and neural embedding (right) strategies. The coverage ratio is very good, being 0.927 in the worst case ($n = 200$ in SNOMED-NCI) and 0.99 in the best case ($n = 2$ in FMA-NCI). This means that, in the worst case, almost 93% of the available reference mappings are *covered* by the matching subtasks in $\mathcal{D}^n_{\mathcal{MT}}$. The differences in terms of coverage between the naive and neural embedding strategies are minimal, with the neural embedding strategy providing slightly better results on average. These results reinforce Hypothesis 1 as the coverage with respect to system-generated mappings is expected to be even better.

---

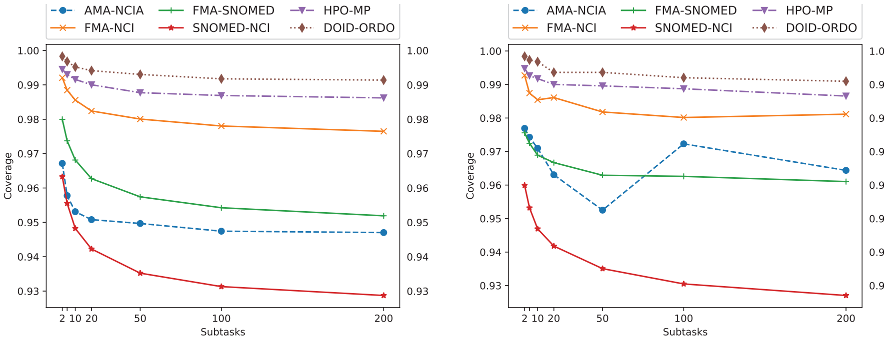
[6] StarSpace: `https://github.com/facebookresearch/StarSpace`

[7] Note that in the context of neural embedding models the term entity refers to objects of different kind, *e.g.*, a word, a sentence, a document or even an ontology entity.

[8] Java codes: `https://github.com/ernestojimenezruiz/logmap-matcher`

[9] Python codes: `https://github.com/plumdeq/neuro-onto-part`

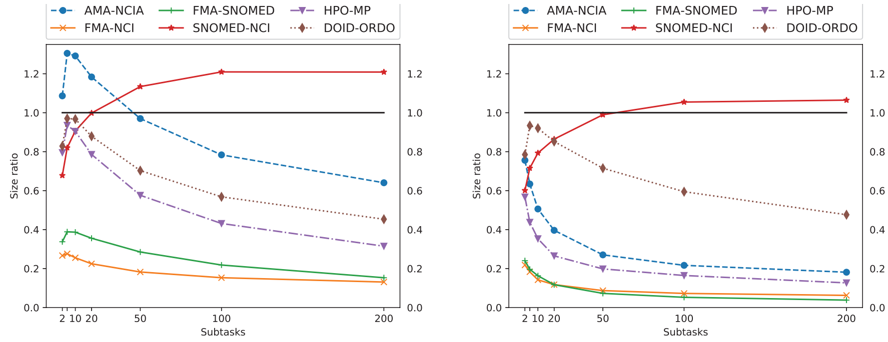
[10] Extended evaluation material in [1] and `https://doi.org/10.5281/zenodo.1214149`



(a) Naive strategy

(b) Neural embedding strategy

Fig. 2: CoverageRatio of $\mathcal{D}^n_{\mathcal{MT}}$ with respect to the number of matching subtasks n.



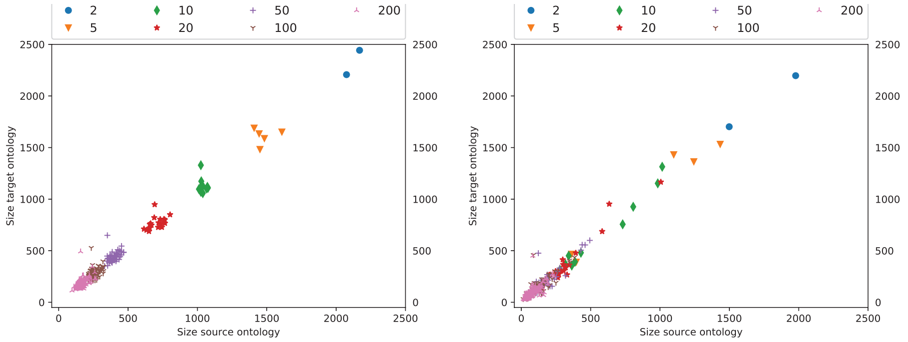
(a) Naive strategy

(b) Neural embedding strategy

Fig. 3: SizeRatio of $\mathcal{D}^n_{\mathcal{MT}}$ with respect to the number of matching subtasks n.

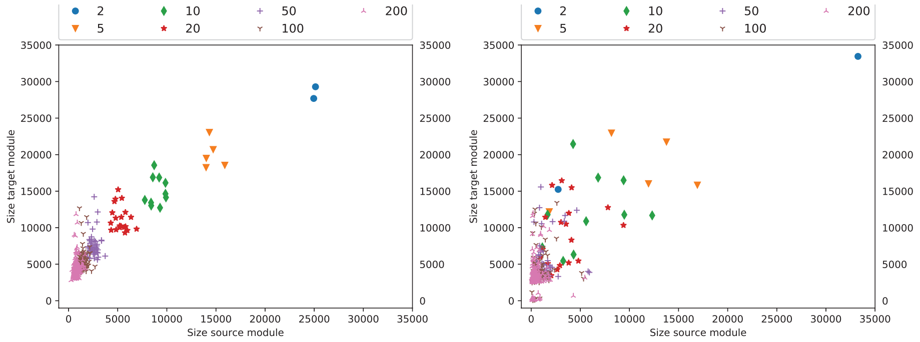
Size ratio. The results in terms of the size (*i.e.*, search space) of the selected divisions $\mathcal{D}^n_{\mathcal{MT}}$ are presented in Figure 3 for the naive (left) and neural embedding (right) strategies. The results with the neural embedding strategy are extremely positive, while the results of the naive strategy, although slightly worse as expected, are surprisingly very competitive. Both strategies improve the search space with respect to the original $\mathcal{MT}$ for all cases with the exception of the naive strategy in the AMA-NCIA case with $n < 50$, and the SNOMED-NCI case with $n > 20$, which validates Hypothesis 3. SNOMED-NCI confirms to be the hardest case in the *largebio* track. Here the size ratio increases with the number of matching subtasks n and gets stable with $n > 100$.

Size of the source and target modules. The scatter plots in Figures 4 and 5 visualize the size of the source modules against the size of the target modules for the matching tasks in each division $\mathcal{D}^n_{\mathcal{MT}}$. For instance, the (orange) triangles represent points $\big(|Sig(\mathcal{O}_1^i)|, |Sig(\mathcal{O}_2^i)|\big)$ being $\mathcal{O}_1^i$ and $\mathcal{O}_2^i$ the source and target modules (with i=1,...,5) in the matching subtasks of $\mathcal{D}^5_{\mathcal{MT}}$. Figure 4 shows the plots for the AMA-NCIA case while Figure 5 for the FMA-NCI case, using the naive (left) and neural embedding (right) strategies. The naive strategy leads to rather balanced an similar tasks (note differentiated cloud of points) for each division $\mathcal{D}^n_{\mathcal{MT}}$ for both cases. The neural embedding strategy has more variability in the size of the tasks within a given division $\mathcal{D}^n_{\mathcal{MT}}$. In the FMA-NCI case the tasks generated by the neural embedding strategy are also less balanced and the target module tends to be larger than the source mod-



(a) Naive strategy (b) Neural embedding strategy

Fig. 4: Source and target module sizes in the computed subtasks for AMA-NCIA.



(a) Naive strategy (b) Neural embedding strategy

Fig. 5: Source and target module sizes in the computed subtasks for FMA-NCI.

ule. Nonetheless, on average, the (aggregated) size of the matching tasks in the neural embedding strategy are significantly reduced as shown in Figure 3.

Computation times. The time to compute the divisions of the matching task is tied to the number of locality modules to extract, which can be computed in polynomial time relative to the size of the input ontology [10]. The creation of LexI does not add an important overhead, while the training of the neural embedding model in the advance strategy ranges from 21s in AMA-NCI to 224s in SNOMED-NCI. Overall, for example, the required time to compute the division with 50 matching subtasks ranges from 2s in AMA-NCIA to 413s in SNOMED-NCI with the naive strategy, and from 24s (AMA-NCIA) to 647s (SNOMED-NCI) with the neural embedding strategy.

4.2 Evaluation of OAEI systems

In this section we support Hypothesis 2 by showing that the division of the alignment task enables systems that, given some computational constraints, were unable to complete an OAEI task. We have selected the following five systems from the latest OAEI campaigns: MAMBA [5], GMap [13], FCA-Map [6], KEPLER [14], and POMap [7]. MAMBA and GMap failed to complete the OAEI 2015 Anatomy track [2] with 8Gb of allocated memory, while FCA-Map, KEPLER and POMap could not complete the largest tasks in the *largebio* track within a 12 hours time-frame (with 16Gb of allocated

Table 3: Evaluation of systems that failed to complete OAEI tasks in the 2015-2017 campaigns. (*) GMap was tested allocating 8Gb of memory. Time reported in hours (h).

Tool	**Task**	**Year**	**Matching subtasks**	**Naive strategy**				**Neural embedding strategy**			
				P	**R**	**F**	**t (h)**	**P**	**R**	**F**	**t (h)**
GMap (*)	Anatomy	2015	5	0.87	0.81	0.84	1.3	0.88	0.82	0.85	0.7
			10	0.85	0.81	0.83	1.7	0.86	0.82	0.84	0.8
MAMBA	Anatomy	2015	20	0.88	0.63	0.73	2.3	0.89	0.62	0.73	1.0
			50	0.88	0.62	0.73	2.4	0.89	0.62	0.73	1.0
FCA-Map	FMA-NCI	2016	20	0.56	0.90	0.72	4.4	0.62	0.90	0.73	3.1
			50	0.58	0.90	0.70	4.1	0.60	0.90	0.72	3.0
KEPLER	FMA-NCI	2017	20	0.45	0.82	0.58	8.9	0.48	0.80	0.60	4.3
			50	0.42	0.83	0.56	6.9	0.46	0.80	0.59	3.8
POMap	FMA-NCI	2017	20	0.54	0.83	0.66	11.9	0.56	0.79	0.66	5.7
			50	0.55	0.83	0.66	8.8	0.57	0.79	0.66	4.1

memory) [2].[11] Note that GMap and MAMBA were also tested in the OAEI 2015 with 14Gb of memory. This new setting allowed GMap to complete the task [2].

Table 3 shows the obtained results in terms of computation times, precision, recall and f-measure over different divisions $\mathcal{D}^n_{\mathcal{MT}}$ computed by the naive and neural embedding strategies. For example, MAMBA was run over divisions with 20 and 50 matching subtasks (*i.e.*, $n \in \{20, 50\}$). Note that GMap was tested allocating only 8Gb of memory as with this constraint it could not complete the task in the OAEI 2015. The results can be summarized as follows:

i) The computation times are encouraging since the (independent) matching tasks have been run sequentially without any type of parallelization.
ii) Times also include loading the ontologies from disk for each matching task. This step could be avoided if subtasks are directly provided by the presented framework.
iii) We did not perform an exhaustive analysis, but memory consumption was lower than 8Gb in all tests; thus, systems like GMap could run under limited resources.
iv) The increase of matching subtasks is beneficial for FCA-Map, KEPLER and POMap in terms of computation times. This is not the case for MAMBA and GMap.
v) The division generated by the neural embedding strategy leads to smaller computation times than the naive strategy counterparts, as expected from Figure 3.
vi) The f-measure is slightly reduced as the size of n increases.

Comparison with OAEI results. There are *baseline* results in the OAEI for the selected systems [2], with the exception of MAMBA where the results are novel for the *anatomy* track. GMap, if 14Gb were allocated, was able to complete the *anatomy* task and obtained an f-measure of 0.861. KEPLER, POMap and FCA-Map completed the OAEI task involving small fragments of FMA-NCI (*i.e.*, the *overlapping matching* task as in Definition 4) with an f-measure of 0.891, 0.861 and 0.935, respectively. The f-measure using the divisions of the matching task is slightly lower for GMap. The results are much lower for the cases of KEPLER, POMap and FCA-Map, but they cannot be fully comparable as systems typically reduce their performance when dealing with the whole *largebio* ontologies [2]. The authors of FCA-Map have also recently reported results for an improved version of FCA-Map [15]. They completed the FMA-NCI task in near

---

[11] In a preliminary evaluation round a 4 hours time-frame was given, which was later extended.

7 hours, with a precision of 0.41, a recall of 0.87 and a f-measure of 0.56. The results obtained with $\mathcal{D}^{20}_{\mathcal{MT}}$ and $\mathcal{D}^{50}_{\mathcal{MT}}$ are thus very positive, since both strategies lead to much better numbers in terms of computation times and f-measure.

5 Related work

Partitioning has been widely used to reduce the complexity of the ontology alignment task. In the literature there are two major categories of partitioning techniques, namely: *independent* and *dependent*. Independent techniques typically use only the structure of the ontologies and are not concerned about the ontology alignment task when performing the partitioning. Whereas dependent partitioning methods rely on both the structure of the ontology and the ontology alignment task at hand. Although our approach does not compute (non-overlapping) partitions of the ontologies, it can be considered a dependent technique.

Prominent examples of ontology alignment systems including partitioning techniques are Falcon-AO [16], COMA++ [17] and TaxoMap [18]. COMA++ and Falcon-AO perform independent partitioning where the clusters of the source and target ontologies are independently extracted. Then pairs of similar clusters (*i.e.*, matching subtasks) are aligned using standard techniques. TaxoMap [18] implements a dependent technique where the partitioning is combined with the matching process. TaxoMap proposes two methods, namely: PAP (partition, anchor, partition) and APP (anchor, partition, partition). The main difference of these methods is the order of extraction of (preliminary) anchors to discover pairs of partitions to be matched (*i.e.*, matching subtasks).

The above approaches, although they present interesting results, did not provide any guarantees about the coverage (as in Definition 2) of the discovered partitions. In [19] we performed a preliminary study with the PBM method of Falcon-OA, and the PAP and APP methods of TaxoMap. The results in terms of coverage with the *largebio* tasks were very low, which directly affected the results of the evaluated systems. These rather negative results encouraged us to work on the approach presented in this paper.

Our dependent approach, unlike traditional partitioning methods, computes overlapping self-contained modules (*i.e.*, locality modules). Locality modules guarantee the extraction of all semantically related entities for a given signature, which enhances the coverage results and enables the inclusion of the relevant information required by an alignment system. It is worth mentioning that the need of self-contained and covering modules was also highlighted in a preliminary work by Paulheim [20].

6 Conclusions and future work

We have developed a novel framework to split the ontology alignment task into several matching subtasks based on a lexical index and locality modules. We have also presented two clustering strategies of the lexical index. One of them relies on a simple splitting method, while the other relies on a fast (log-linear) neural embedding model. We have performed a comprehensive evaluation of both strategies. The achieved high coverage (*i.e.*, minimal information loss) in combination with the reduction of the search space and the small computation times suggests that the computed divisions based on LexI are suitable in practice. The division of the matching task allowed us to obtain results for five systems which failed to complete these OAEI matching tasks in the past.

Both the naive and the neural embedding strategies require the size of the number of matching subtasks or clusters as input. The (required) matching subtasks may be known before hand if, for example, the matching tasks are to be run in parallel in a number of available CPUs. For the cases where the resources are limited or where a matching system is known to cope with small ontologies, we plan to design an algorithm to estimate the number of clusters so that the size of the matching subtasks in the computed divisions is appropriate to the system and resource constraints.

As immediate future we plan to extend the conducted evaluation to better understand the impact of the division over different ontology alignment systems. We also aim at studying different notions of *context* tailored to the ontology alignment task.

Acknowledgements. EJR was funded by the Centre for Scalable Data Access (SIRIUS), the RCN project BigMed, and The Alan Turing project AIDA.

References

1. Jimenez-Ruiz, E., Agibetov, A., Samwald, M., Cross, V.: Breaking-down the Ontology Alignment Task with a Lexical Index and Neural Embeddings. arXiv (2018) Available from: `https://arxiv.org/abs/1805.12402`.
2. (Ontology Alignment Evaluation Initiative) : `http://oaei.ontologymatching.org/`.
3. Ngo, D., Bellahsene, Z., Coletta, R.: YAM++ results for OAEI 2011. In: 6th International Workshop on Ontology Matching. (2011)
4. Huber, J., Sztyler, T., Nößner, J., Meilicke, C.: CODI: combinatorial optimization for data integration: results for OAEI 2011. In: 6th Int'l Workshop on Ontology Matching. (2011)
5. Meilicke, C.: MAMBA - results for the OAEI 2015. In: 10th International Workshop on Ontology Matching. (2015) 181–184
6. Zhao, M., Zhang, S.: FCA-Map results for OAEI 2016. In: 11th International Workshop on Ontology Matching. (2016)
7. Laadhar, A., Ghozzi, F., Megdiche, I., Ravat, F., Teste, O., Gargouri, F.: POMap results for OAEI 2017. In: 12th International Workshop on Ontology Matching. (2017) 171–177
8. Jiménez-Ruiz, E., Cuenca Grau, B., Zhou, Y., Horrocks, I.: Large-scale interactive ontology matching: Algorithms and implementation. In: European Conf. Artif. Intell. (ECAI). (2012)
9. Wu, L., Fisch, A., Chopra, S., Adams, K., Bordes, A., Weston, J.: StarSpace: Embed All The Things! arXiv (2017)
10. Cuenca Grau, B., Horrocks, I., Kazakov, Y., Sattler, U.: Modular reuse of ontologies: Theory and practice. J. Artif. Intell. Res. **31** (2008) 273–318
11. Euzenat, J., Shvaiko, P.: Ontology Matching, Second Edition. Springer (2013)
12. Harrow, I., Jiménez-Ruiz, E., et al.: Matching disease and phenotype ontologies in the ontology alignment evaluation initiative. J. Biomedical Semantics **8**(1) (2017)
13. Li, W., Sun, Q.: GMap: results for OAEI 2015. In: 10th International Workshop on Ontology Matching. (2015) 150–157
14. Kachroudi, M., Diallo, G., Yahia, S.B.: OAEI 2017 results of KEPLER. In: 12th International Workshop on Ontology Matching. (2017) 138–145
15. Zhao, M., Zhang, S., Li, W., Chen, G.: Matching biomedical ontologies based on formal concept analysis. J. Biomedical Semantics **9**(1) (2018) 11:1–11:27
16. Hu, W., Qu, Y., Cheng, G.: Matching large ontologies: A divide-and-conquer approach. Data Knowl. Eng. **67** (2008) 140–160
17. Algergawy, A., Massmann, S., Rahm, E.: A clustering-based approach for large-scale ontology matching. In: ADBIS. (2011) 415–428
18. Hamdi, F., Safar, B., Reynaud, C., Zargayouna, H.: Alignment-based partitioning of large-scale ontologies. In: Advances in Knowledge Discovery and Management. (2009) 251–269
19. Pereira, S., Cross, V., Jiménez-Ruiz, E.: On partitioning for ontology alignment. In: International Semantic Web Conference (Posters & Demonstrations). (2017)
20. Paulheim, H.: On Applying Matching Tools to Large-scale Ontologies. In: OM. (2008)

Interactive Ontology Matching: Using Expert Feedback to Select Attribute Mappings

Jomar da Silva[1], Kate Revoredo[1], Fernanda Araujo Baião[1], and Jérôme Euzenat[2]

[1] Department of Applied Informatics
Federal University of the State of Rio de Janeiro (UNIRIO), Brazil
{jomar.silva, katerevoredo, fernanda.baiao}@uniriotec.br
[2] Univ. Grenoble Alpes, Inria, CNRS, Grenoble INP, LIG, F-38000 Grenoble France
Jerome.Euzenat@inria.fr

Abstract. Interactive Ontology Matching considers the participation of domain experts during the matching process of two ontologies. An important step of this process is the selection of mappings to submit to the expert. These mappings can be between concepts, attributes or relationships of the ontologies. Existing approaches define the set of mapping suggestions only in the beginning of the process before expert involvement. In previous work, we proposed an approach to refine the set of mapping suggestions after each expert feedback, benefiting from the expert feedback to form a set of mapping suggestions of better quality. In this approach, only concept mappings were considered during the refinement. In this paper, we show a new approach to evaluate the benefit of also considering attribute mappings during the interactive phase of the process. The approach was evaluated using the OAEI conference data set, which showed an increase in recall without sacrificing precision. The approach was compared with the state-of-the-art, showing that the approach has generated alignment with state-of-the-art quality.

Keywords: ontology matching, Wordnet, interactive ontology matching, ontology alignment, interactive ontology alignment

1 Introduction

Ontology matching aims to discover correspondences (mappings) between entities of different ontologies [1]. One of its strategies is the interactive one. Interactive ontology matching approaches consider the knowledge of domain experts during the matching process. The interaction with the user can be used to improve the results over fully automatic approaches [2]. An important step of this strategy is the definition of the set of mappings to be submitted to the expert for feedback. This set to be submitted to the expert was called, in this paper, set of mapping suggestions. Existing approaches [3][4][5][6][7][8][9][10][11][12][13][14][15] define this set before the interaction with the expert begins; thus, the approaches do not use expert feedback to select mappings to the set of mapping suggestions.

In previous work [16], we combined a structural and a semantic technique for interactively considering the expert feedback in the revision of the set of mapping suggestions, but taking into account only concept mappings. However, considering also the properties of these concepts may bring a better integration of the ontologies.

In this work, we propose $ALIN_{Attr}$ to evaluate the benefit of also considering attribute mappings during the interactive strategy. The attribute mappings suggested are associated with the concept mappings evaluated by the expert; therefore, they are more prone to be correct and potentially increase the recall compared with existing strategies that automatically include attribute mappings [14][15].

The evaluation results evidenced the benefit of considering attributes during the interactive phase, using a heuristic for choosing the attribute mappings inspired on the Stable Marriage Problem [17][18]. In addition, the current approach was compared to the state-of-the-art.

The rest of this paper is organized as follows. Section 2 reviews interactive ontology matching. Section 3 presents the approach, which is called $ALIN_{Attr}$, and its implementation. Section 4 describes our evaluation methodology and discusses experimental results. Finally, section 5 concludes the paper.

2 Interactive Ontology Matching

An interactive ontology matching process is an ontology matching process considering the involvement of domain experts. In this paper, we consider this involvement as the domain experts providing feedback about mappings of ontologies entities, that is, mapping are presented to the expert who replies which of them should be accepted or rejected. Therefore, the approach takes advantage of the knowledge of domain experts towards finding an alignment.

The most relevant steps in this process are the selection of the mappings to receive expert feedback and the propagation of this feedback. Furthermore, the propagation may also impact the mappings selected for future expert feedback. The different existing approaches for interactive ontology matching vary in techniques for these two steps.

In the selection step, the existing approaches of interactive ontology matching use similarity metrics to select the set of mapping suggestions. The similarity metric is a function that returns a numeric value, indicating the similarity between the two entities of a mapping, according to some criterion. An approach can associate one or several similarity values, each of a different similarity metric, to a mapping.

In the selection step, the approaches can use multiple matchers, algorithms that receive, as input, entities and generate, as output, mappings. Each matcher can use different similarity metrics, among other features. At the end of the selection step, the results of these matchers can be combined and filtered generating the set of mapping suggestions [13].

In the propagation step, user feedback can be used in different ways. Some approaches automatically classify some mapping suggestions using a threshold, a value that indicates whether a mapping should be automatically accepted (in some cases rejected) if its similarity values are greater (or smaller) than it. Expert feedbacks are used to calculate this threshold [3][4][5][6][7]. Some approaches automatically classify some mappings of the set of mapping suggestions using a classifier. These approaches use expert feedbacks to create the training dataset for learning the classifiers [8][9]. Some approaches use expert feedbacks to modify the weight of similarity metrics [5][6][10] or to directly change the value of similarity metrics [11][12]. Expert feedbacks are also used to remove mapping suggestions from the set of mapping suggestions [13][14][15].

3 The $ALIN_{Attr}$ Approach

In this section, we describe our approach, $ALIN_{Attr}$, for interactively matching two ontologies. $ALIN_{Attr}$, at each interaction, uses expert feedback to remove mapping suggestions and include new attribute mapping suggestions into the set of mapping suggestions.

The $ALIN_{Attr}$ *top-level algorithm* (Algorithm 1) starts with a pair of ontologies (O and O') and a set of similarity metrics (SoM). Then, it splits in two main steps. The first one defines the initial mapping suggestions (SMS) and the initial alignment (A) (line 1 to line 17 of Algorithm 1) and the second one interactively receives expert feedback to a mapping suggestion and propagate it (lines 18 to 29 of Algorithm 1).

The initialization step starts collecting all concepts of ontology O (SCO) and O' (SCO') and then for each similarity metric (SimM) a set of mapping suggestions is found using the *simple matching algorithm* (line 5 of Algorithm 1). This algorithm treats the matching problem as a stable marriage problem with size list limited to 1 [17][18], i.e., the algorithm only selects one mapping if similarity value between the two entities of the mapping is the highest considering all the mappings with at least one of these entities (Algorithm 2). At this moment only concept mappings, not property mappings, are chosen. The initial set of mapping suggestions is defined as the union of the mapping suggestions found for each similarity metric (lines 6 to 10 of Algorithm 1). The mappings in which their entity names are the same are placed in the alignment and removed from the set of mapping suggestions (lines 12 to 17 of Algorithm 1). Moreover, $ALIN_{Attr}$ inserts into the set of mapping suggestions attribute mappings associated with these concept mappings placed in the alignment (line 15 of Algorithm 1). The approach uses the *structural attribute selection technique*, which will be explained later, to choose the attribute mappings.

After defining the initial set of mapping suggestions and the initial alignment, $ALIN_{Attr}$ moves to the interactive step, in which the mapping suggestions receive the feedback of the expert (line 20 of Algorithm 1). If the expert accepts a mapping suggestion, then it is included in the alignment (line 23 of Algorithm

Algorithm 1 $ALIN_{Attr}$ Top-level Algorithm

```
Input: $O$, $O'$, SoM
Output: A
        /*Initialization step*/
 1: A = $\emptyset$; SMS = $\emptyset$;
 2: $SCO \leftarrow$ all concepts of $O$;
 3: $SCO' \leftarrow$ all concepts of $O'$;
 4: for each SimM $\in$ SoM do
 5:     M $\leftarrow$ Simple Matching Algorithm($SCO$,$SCO'$,SimM);
 6:     for each $m(e,e') \in$ M do
 7:         if $m(e,e') \notin$ SMS then
 8:             add $m(e,e')$ to SMS;
 9:         end if
10:     end for
11: end for
12: for each $m(e,e') \in$ SMS do
13:     if name of $e$ = name of $e'$ then
14:         move $m(e,e')$ from SMS to A;
15:         SMS $\leftarrow$ Structural Attribute Selection Technique($m(e,e')$,SoM);
16:     end if
17: end for
        /*Interactive step*/
18: while SMS $\neq \emptyset$ do
19:     select $m(e,e') \in$ SMS with the biggest sum of similarity metrics;
20:     receive expert feedback on $m(e,e')$;
21:     remove $m(e,e')$ from SMS;
22:     if $m(e,e')$ is accepted then
23:         add $m(e,e')$ to A;
24:         SMS $\leftarrow$ Remove Mappings with Equal Entities(SMS,$m(e,e')$);
25:         if $m(e,e')$ is a concept mapping then
26:             SMS $\leftarrow$ Structural Attribute Selection Technique($m(e,e')$,SoM);
27:         end if
28:     end if
29: end while
30: return A
```

1). $ALIN_{Attr}$ simulates the expert feedback by accessing a reference alignment. Session 4 further explains the reference alignment.

Up to this point, as we use several similarity metrics and the set of mapping suggestions is the union of the formed sets made for each metric there may be mappings with one of the entities equal. Since we want to generate a one-to-one alignment, once one of these mappings is accepted, the others will be rejected and removed from the set of mapping suggestions (line 24 of Algorithm 1) It is worth noting that $ALIN_{Attr}$ uses expert feedback to reject these mappings. If

Algorithm 2 Simple Matching Algorithm

```
Input: $SE$, $SE'$, SimM
Output: M
for each $e \in SE$ do
    $max_{e'} \leftarrow \max_{e' \in SE'} SimM(e, e')$;
    $max_e \leftarrow \max_{e'' \in SE} SimM(e'', max_{e'})$;
    if $e = max_e$ then
        add $m(e, max_{e'})$ to M;
    end if
end for
return M;
```

$ALIN_{Attr}$ would automatically reject these mappings, it would probably make mistakes.

At this point, the $ALIN_{Attr}$ approach uses the *structural attribute selection technique* which will try to select, based on expert feedback, the best attribute mappings to be included into the set of mapping suggestions. The assumption behind the *structural attribute selection technique* is that if the attributes in an attribute mapping are attributes of concepts of a concept mapping, then this attribute mapping is more likely to be correct.

Algorithm 3 describes the *structural attribute selection technique*. It considers all attributes of the concepts of the input accepted mapping (lines 1 and 2 of Algorithm 3) and for each similarity metric it uses the *simple matching algorithm* to define attribute mapping suggestions. The output of the algorithm is the union of the set of attribute mappings found for each similarity metric.

Algorithm 3 Structural Attribute Selection Technique

```
Input: $m(c, c')$,SoM
Output: SMS
 1: $SA \leftarrow$ all attributes of $c$;
 2: $SA' \leftarrow$ all attributes of $c'$;
 3: for each $SimM \in SoM$ do
 4:     M $\leftarrow$ $Simple\ Matching\ Algorithm(SA, SA', \mathrm{SimM})$;
 5:     for each $m(a, a') \in$ M do
 6:         if $m(a, a') \notin$ SMS then
 7:             add $m(a, a')$ to SMS;
 8:         end if
 9:     end for
10: end for
11: return SMS
```

Instead of selecting mappings between concepts of the two ontologies, like in the $ALIN_{Attr}$ *top-level algorithm*, the *structural attribute selection technique* (Algorithm 3) uses the *simple matching algorithm* (Algorithm 2) to select mappings between attributes of the concepts in an accepted mapping. The use of the *simple matching algorithm* proved to be efficient in choosing the attribute mappings to be inserted in the set of mapping suggestions, as will be shown later in this paper.

$ALIN_{Attr}$ was implemented in Java using the following Java APIs: Stanford coreNLP API [19] with a routine to put a word in canonical form; Simmetrics API [20], with string-based similarity metrics; HESML API [21], with Wordnet [22] based linguistic metrics; And the Alignment API [23], which contains routines for handling ontologies written in OWL. The most frequent synsets of words are used to calculate semantic similarities. To find this synset is used the WS4J API[3].

4 Experimental Evaluation

In this section, we evaluate our approach for interactive ontology matching considering attribute mappings.

4.1 Configuration of the experiment

The evaluation is designed towards answering three research questions:

> RQ1: Does the consideration of attribute mappings improve the quality of the final alignment?
>
> RQ2: Does the use of expert feedback for the inclusion of attribute mappings in the set of mapping suggestions improve the quality of the final alignment?
>
> RQ3: Does the *simple matching algorithm* between the attributes of the concepts improve the quality of the final alignment?

The quality of an alignment is generally measured by F-measure, which is the harmonic mean between recall and precision. In an interactive approach another quality metric should be taken into account, the number of interactions with the expert that was necessary to achieve the alignment. The lower the number of interactions, the better. Thus, the two quality metrics were used to answer the research questions in this work.

---

[3] 'WS4J'. Available at https://github.com/Sciss/ws4j Last accessed on Jan, 16, 2018.

Algorithm 4 Attribute Inclusion Technique for $ALIN_{AttrAuto}$

```
Input: $O$, $O'$, SoM
Output: SMS
$SA \leftarrow$ all attributes of $O$;
$SA' \leftarrow$ all attributes of $O'$;
for $SimM \in SoM$ do
    M $\leftarrow$ Simple Matching Algorithm($SA$,$SA'$,SimM);
    for each $m(e, e') \in$ M do
        if $m(e, e') \notin$ SMS then
            add $m(e, e')$ to SMS;
        end if
    end for
end for
return SMS
```

Algorithm 5 Structural Attribute Selection Technique for $ALIN_{AttrFBack}$

```
Input: $m(c, c')$,SoM
Output: SMS
$SA \leftarrow$ all attributes of c;
$SA' \leftarrow$ all attributes of c';
for each $a \in SA$ do
    for each $a'$ in $\in SA'$ do
        add $m(a, a')$ to SMS;
    end for
end for
return SMS
```

Towards answering these questions, some variations of $ALIN_{Attr}$ were considered:

- $ALIN_{WAttr}$: This variation didn't take into account attribute mappings, i.e., only concept mappings compose the set of mapping suggestions. For that, the $ALIN_{WAttr}$ variation removes the calls for the *structural attribute selection technique* (Algorithm 3) in line 15 and from line 25 to line 27 of the $ALIN_{Attr}$ *top-level algorithm* (Algorithm 1).
- $ALIN_{AttrAuto}$: This variation includes the attribute mappings only in the initialization step, i.e., not considering expert feedback. For that, the $ALIN_{AttrAuto}$ variation removes the calls for the *structural attribute selection technique* (Algorithm 3) in line 15 and from line 25 to line 27 in the $ALIN_{Attr}$ *top-level algorithm* (Algorithm 1) and includes a call for *attribute inclusion technique for* $ALIN_{AttrAuto}$ (Algorithm 4) in the $ALIN_{Attr}$ *top-level algorithm* (Algorithm 1) after line 17.
- $ALIN_{AttrFBack}$: This variation includes all attribute mappings related to the accepted concept mapping into the set of mapping suggestions, i.e., this

variation doesn't use the *simple matching algorithm* (Algorithm 2) to reduce the number of included attribute mappings. For that, the $ALIN_{AttrAuto}$ variation makes a call to the *structural attribute selection technique for* $ALIN_{AttrFBack}$ (Algorithm 5) instead of a call to the *structural attribute selection technique* (Algorithm 3) in lines 15 and 26 of the $ALIN_{Attr}$ *top-level algorithm* (Algorithm 1).

OAEI provides several data sets, which are sets of ontologies, to be used in the evaluation of ontology matching tools. From the data sets provided by OAEI, the only one that contained documentation of attributes and that had size that allowed the execution of $ALIN_{Attr}$ is the conference data set. Therefore, the conference data set was used to evaluate the approach. OAEI provides reference alignments, which are alignments that contains the mappings that are believed to be correct, between the pairs of the ontologies of the conference data set. In the $ALIN_{Attr}$ approach, a reference alignment query simulates the consult to the expert. The selection of the similarity metrics was based on two criteria: available implementations and the result of these metrics in assessments, such as those carried out in [24] and [25]. Based on [24] and [25], $ALIN_{Attr}$ uses Jaccard, Jaro-Wrinkler and n-gram string-based metrics and the Resnick, Jiang-Conrath and Lin linguistic metrics. Resnick, Jiang-Conrath and Lin are metrics that require a taxonomy to be computed [24], this taxonomy being provided, in this algorithm, by Wordnet [22].

4.2 Results

The results in terms of number of interactions (NI), precision, recall and F-measure can be seen in Table 1.

Table 1. Comparison between different $ALIN_{Attr}$ variations executions with Conference Data Set

	Total of questions	NI	Precision	F-measure	Recall
$ALIN_{WAttr}$	1183	582	0.921	0.783	0.692
$ALIN_{AttrAuto}$	1574	739	0.905	0.809	0.741
$ALIN_{AttrFBack}$	1321	631	0.924	0.817	0.741
$ALIN_{Attr}$	1242	614	0.924	0.815	0.738

In each interaction with the expert, up to three mapping suggestions can be presented, since each mapping suggestion has one entity in common with another mapping suggestion of the interaction [26].

Comparing $ALIN_{WAttr}$ with the other three approaches, that considered attributes mappings, we can see the improvement in the recall, which was expected since other mappings were evaluated. It is also possible to notice an increase in the number of interactions with the expert. Therefore, the inclusion of attribute mappings without taking into account the expert feedback generates an increase in the F-measure, but also an increase in the number of interactions with the expert leading to an inconclusive answer to the RQ1 question.

Comparing $ALIN_{AttrAuto}$, which did not take into account the feedback of the expert, with $ALIN_{AttrFBack}$ and $ALIN_{Attr}$, which considered it, we can observe an improvement in the F-measure and a decrease in the number of interactions with the expert. This demonstrates that using expert feedback is a good practice, answering positively RQ2. It is important to note that it was assumed that the expert did not make mistakes. Therefore, these results are valid when the expert makes no mistakes.

Addressing RQ3, i.e., comparing $ALIN_{Attr}$ with $ALIN_{AttrFBack}$ towards evaluating the benefit of reducing the number attribute mappings by using the *simple matching algorithm*, we observed a decrease in the number of interactions with almost no loss of quality of the alignment, what answer positively to the RQ3 question.

4.3 Comparison between tools that participated in the OAEI interactive conference track

Table 2. Comparison between some the tools of OAEI 2017 Conference Data Set Interactive Tracking and $ALIN_{Attr}$ and $ALIN_{Attr+Syn}$ with 100% hit rate

	Number of questions	NI	Precision	F-measure	Recall
$ALIN_{Attr}$	1242	614	0.924	0.815	0.738
AML [14][27]	270	271	0.912	0.799	0.711
LogMap [15][28]	142	82	0.886	0.723	0.610
XMap [29][30]	4	4	0.837	0.678	0.57
$ALIN_{Attr+Syn}$	443	205	0.918	0.782	0.692

OAEI annually provides a comparison between ontology matching tool performances, and one ontology group used is the conference dataset, used in this paper [31]. Table 2 depicts a comparison between some the tools that participated in the OAEI 2017 interactive conference track and $ALIN_{Attr}$ and $ALIN_{Attr+Syn}$.

The tools AML, LogMap, and XMAP (Table 2) are interactive ontology matching tools. This tools, like $ALIN_{Attr}$, include attribute mappings in the generated alignment but this inclusion is done in a non-interactive way, not taking into account the expert feedback.

The Table 2 depicts results with the expert hitting 100% of the answers. The results showed that $ALIN_{Attr}$ generated a high level result when running the conference data set when the expert hit 100% of the answers, but with a very large number of interactions when compared to the other tools.

To verify the quality of $ALIN_{Attr}$ if it uses a number of interactions more compatible with the other tools, two techniques, described in [16], were added to

$ALIN_{Attr}$. In [16], these techniques proved to be very efficient in reducing the number of interactions without significantly reducing quality. The inclusion of the two techniques generates the results shown on line '$ALIN_{Attr+Syn}$' of Table 2 and shows that, as the quality as the number of interactions, $ALIN_{Attr+Syn}$ is good when compared to other tools.

5 Conclusion

Ontology matching is a necessary step for establishing interoperation among semantic web applications. Its purpose is to discover mappings between the entities of at least two ontologies. The quality of an alignment generated by a matching approach is generally measured by F-measure, which is the harmonic means between recall and precision. Another quality metric, when the ontology matching process is interactive, is the number of interactions with the expert.

An important step in the process of interactive ontology matching is the definition of the set of mapping suggestions, that is, the set of mappings that will be shown to the expert. The problem seen in this paper is how to efficiently include attribute mappings into the set of mapping suggestions. The $ALIN_{Attr}$ approach includes attribute mappings taking advantage of the expert feedback, of the structures of the involved ontologies, as well as the use of the *simple matching algorithm*. Experimental results showed the benefit of the approach when assuming that the expert does not make mistakes.

In addition, the quality of the alignment provided by $ALIN_{Attr}$ was compared to state of the art tools that have participated in the track of interactive ontology matching in OAEI 2017. The results obtained show that $ALIN_{Attr}$ generates an alignment with a good quality in comparison to other tools, with regard to precision, recall and F-measure, when the expert never makes mistakes, but with a number of interactions far superior to other tools. When performed with techniques to decrease the number of interactions, the number of interactions was compatible with that of the other tools, preserving a good quality.

As future work, one interesting direction is to explore how to reduce the negative effects of expert mistakes. The $ALIN_{Attr}$ generates good results when the expert does not make mistakes, but because the approach uses the expert feedback as the input of the structural attribute selection technique, probably incorrect attribute mappings will be generated when the expert makes a mistake.

Acknowledgement

The second author was partially funding by project PQ-UNIRIO Nº01/2017 ("Aprendendo, adaptando e alinhando ontologias:metodologias e algoritmos.") and CAPES/PROAP. The fourth author was partially funding by 'CNPq Special visiting researcher grant (314782/2014-1)'.

References

1. Euzenat, J., Shvaiko, P.: Ontology Matching - Second Edition. Springer-Verlag (2013)
2. Paulheim, H., Hertling, S., Ritze, D.: Towards Evaluating Interactive Ontology Matching Tools. Lecture Notes in Computer Science **7882** (2013) 31–45
3. Paulheim, H., Hertling, S.: Wesee-match results for oaei 2013. In: Proceedings of the 8th International Conference on Ontology Matching - Volume 1111. OM'13, Aachen, Germany, Germany, CEUR-WS.org (2013) 197–202
4. Hertling, S.: Hertuda results for oeai 2012. In: Proceedings of the 7th International Conference on Ontology Matching - Volume 946. OM'12, Aachen, Germany, Germany, CEUR-WS.org (2012) 141–144
5. Duan, S., Fokoue, A., Srinivas, K.: One size does not fit all: Customizing ontology alignment using user feedback. In Patel-Schneider, P.F., Pan, Y., Hitzler, P., Mika, P., Zhang, L., Pan, J.Z., Horrocks, I., Glimm, B., eds.: The Semantic Web – ISWC 2010, Berlin, Heidelberg, Springer Berlin Heidelberg (2010) 177–192
6. Shi, F., Li, J., Tang, J., Xie, G., Li, H.: Actively learning ontology matching via user interaction. In Bernstein, A., Karger, D.R., Heath, T., Feigenbaum, L., Maynard, D., Motta, E., Thirunarayan, K., eds.: The Semantic Web - ISWC 2009, Berlin, Heidelberg, Springer Berlin Heidelberg (2009) 585–600
7. Chunhua Li, Zhiming Cui, P.Z.J.W.J.X.T.H.: Improving ontology matching with propagation strategy and user feedback. In: Seventh International Conference on Digital Image Processing (ICDIP 2015). Volume 9631. (2015) 6
8. Lopes, V., Baião, F., Revoredo, K.: Alinhamento Interativo de Ontologias Uma Abordagem Baseada em Query-by-Committee. Master's thesis, UNIRIO (2015)
9. To H., I.R., Le, H.: An Adaptive Machine Learning Framework with User Interaction for Ontology Matching. Twenty-first International Joint Conference on Artificial Intelligence (2009)
10. Balasubramani, B., Taheri, A., Cruz, I.: User involvement in ontology matching using an online active learning approach. In: CEUR Workshop Proceedings. Volume 1545., CEUR-WS (2015) 45–49
11. Cruz, I.F., Stroe, C., Palmonari, M.: Interactive user feedback in ontology matching using signature vectors. In: 2012 IEEE 28th International Conference on Data Engineering. (April 2012) 1321–1324
12. Cruz, I.F., Loprete, F., Palmonari, M., Stroe, C., Taheri, A.: Pay-as-you-go multi-user feedback model for ontology matching. In Janowicz, K., Schlobach, S., Lambrix, P., Hyvönen, E., eds.: Knowledge Engineering and Knowledge Management, Cham, Springer International Publishing (2014) 80–96
13. Lambrix, P., Kaliyaperumal, R.: A Session-based Ontology Alignment Approach enabling User Involvement. Semantic Web **1** (2016) 1–28
14. Faria, D., Pesquita, C., Santos, E., Palmonari, M., Cruz, I.F., Couto, F.M.: The agreementmakerlight ontology matching system. In Meersman, R., Panetto, H., Dillon, T., Eder, J., Bellahsene, Z., Ritter, N., De Leenheer, P., Dou, D., eds.: On the Move to Meaningful Internet Systems: OTM 2013 Conferences, Berlin, Heidelberg, Springer Berlin Heidelberg (2013) 527–541
15. Jiménez-Ruiz, E., Grau, B.C., Zhou, Y., Horrocks, I.: Large-scale interactive ontology matching: Algorithms and implementation. In: ECAI 2012 - 20th European Conference on Artificial Intelligence. Volume 242. (2012) 444–449
16. Da Silva, J., Baião, F.A., Revoredo, K., Euzenat, J.: Semantic interactive ontology matching: Synergistic combination of techniques to improve the set of candidate

correspondences. In: OM 2017 - 12th ISWC workshop on ontology matching. Volume 2032. (2017) 13–24

17. Gale, D., Shapley, L.S.: College admissions and the stability of marriage. The American Mathematical Monthly **69**(1) (1962) 9–15
18. Irving, R.W., Manlove, D.F., O'Malley, G.: Stable marriage with ties and bounded length preference lists. Journal of Discrete Algorithms **7**(2) (2009) 213 – 219 Selected papers from the 2nd Algorithms and Complexity in Durham Workshop ACiD 2006.
19. Manning, C.D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S.J., McClosky, D.: The Stanford CoreNLP natural language processing toolkit. In: Association for Computational Linguistics (ACL) System Demonstrations. (2014) 55–60
20. Surhone, L.M., Timpledon, M.T., Marseken, S.F.: SimMetrics. VDM Publishing (2010)
21. Lastra-Daz, J.J., Garca-Serrano, A., Batet, M., Fernndez, M., Chirigati, F.: Hesml. Inf. Syst. **66**(C) (June 2017) 97–118
22. Fellbaum, C., ed.: WordNet: An electronic lexical database. MIT Press (1998)
23. David, J., Euzenat, J., Scharffe, F., Trojahn dos Santos, C.: The alignment api 4.0. Semant. web **2**(1) (January 2011) 3–10
24. Petrakis, E.G.M., Varelas, G., Hliaoutakis, A., Raftopoulou, P.: Design and Evaluation of Semantic Similarity Measures for Concepts Stemming from the Same or Different Ontologies object instrumentality. Proceedings of the 4th Workshop on Multimedia Semantics (WMS) **4** (2006) 233–237
25. Cheatham, M., Hitzler, P.: String similarity metrics for ontology alignment. In: Proceedings of the 12th International Semantic Web Conference - Part II. ISWC '13, New York, NY, USA, Springer-Verlag New York, Inc. (2013) 294–309
26. Faria, D.: Using the SEALS Client's Oracle in Interactive Matching (2016)
27. Faria, D., Balasubramani, B.S., Shivaprabhu, V.R., Mott, I., Pesquita, C., Couto, F.M., Cruz, I.F.: Results of AML in OAEI 2017. In: CEUR Workshop Proceedings. Volume 2032. (2017) 122–128
28. Jimènez-Ruiz, E., Grau, B.C., Cross, V.: LogMap family participation in the OAEI 2017. In: CEUR Workshop Proceedings. Volume 2032. (2017) 153–157
29. Djeddi, W.E., Khadir, M.T.: A novel approach using context-based measure for matching large scale ontologies. In Bellatreche, L., Mohania, M.K., eds.: Data Warehousing and Knowledge Discovery, Cham, Springer International Publishing (2014) 320–331
30. Djeddi, W.E., Khadir, M.T., Yahia, S.B.: XMap : Results for OAEI 2017. In: CEUR Workshop Proceedings. Volume 2032. (2017) 196–200
31. Achichi, M., Cheatham, M., Dragisic, Z., Euzenat, J., Faria, D., Ferrara, A., Flouris, G., Fundulaki, I., Harrow, I., Ivanova, V., Jimenez-Ruiz, E., Kolthoff, K., Kuss, E., Lambrix, P., Leopold, H., Li, H., Meilicke, C., Mohammadi, M., Montanelli, S., Pesquita, C., Saveta, T., Shvaiko, P., Splendiani, A., Stuckenschmidt, H., Thieblin, E., Todorov, K., Trojahn, C., Zamazal, O.: Results of the Ontology Alignment Evaluation Initiative 2017. In: Proceedings of the 12th International Workshop on Ontology Matching co-located with the 16th International Semantic Web Conference (ISWC 2017) Vienna, Austria, October 21st, 2017. (2017)

Ontology Augmentation Through Matching with Web Tables

Oliver Lehmberg[1] and Oktie Hassanzadeh[2]

[1] University of Mannheim, B6 26, 68159 Mannheim, Germany
[2] IBM Research, Yorktown Heights, New York, U.S.A.

Abstract. In this paper, we examine the possibility of using data collected from millions of tables on the Web to extend an ontology with new attributes. There are two major challenges in using such a large number of potentially noisy tables for this task. First, table columns need to be matched to create groups of columns that represent a new (or existing) attribute for a particular class in the ontology. Second, the column groups need to be ranked according to their "usefulness" in augmenting the ontology. We show several approaches to addressing these challenges and report on the results of our extensive experiments using Web Tables from the Web Data Commons corpus, and using the DBpedia Ontology as our target ontology.

1 Introduction

The Web is a vast source of valuable knowledge that can be used to extend or augment a given ontology. Knowledge extraction from the Web is a well-studied problem and an active area of research [5, 7, 11]. While such knowledge is often extracted from textual (or semi-structured) contents using information extraction and wrapper induction techniques, there have also been attempts in using the structured data that is exposed on web pages as HTML tables [5, 14, 13].

In this paper, we examine the possibility of using Web tables to augment a given ontology with a new set of attributes. Our hypothesis is that for each class in the given ontology, there are tables on the Web describing instances of the class and their various attributes. Further, not only a large number of these attributes are not already captured in the ontology, but many are not considered "useful", i.e., may be irrelevant, inaccurate, or redundant.

The approach we take in this work is a two-step process. First, tables are matched among each other and to the target ontology, to group columns that refer to the same attribute and align them with classes and existing attributes in the target ontology. The second step ranks the column groups based on a measure of *quality* or usefulness of the group in augmenting the existing attributes in the target ontology. We perform an empirical study of the performance of this approach in using Web Tables extracted from the Common Crawl[3] to augment the properties in DBpedia ontology.

---

[3] http://commoncrawl.org/

2 Related Work

The pioneering work using Web tables to discover new attributes was done by Cafarella et al. in 2008 [3]. They create the so-called "attribute correlation statistics database (AcsDb)" which contains attribute counts based on the column headers in a large corpus of Web tables. From these counts, they estimate attribute occurrence probabilities. Applications for this database are a schema auto-complete function, synonym generation and a tool enabling easy join graph traversal for end-users. We extend their approach as we use clusters derived from matched columns instead of columns headers as basic unit for the statistics.

Das Sarma et al. [4] use label- and value-based schema matching methods to map Web tables to a given query table. For their "Schema Complement" operation they consider all unmapped columns and rank them using the AcsDb and the entity coverage of the input table provided by the user. Their goal is to rank complete tables by their usefulness for the complement task. While they use a matching of Web table columns to the query table to rule out existing attributes, when it comes to finding new attributes, they fall back to the AcsDb approach. In contrast to that, we calculate attribute statistics based on matched column clusters.

Lee et al. [9] extract attributes from Probase [15], Web documents, a search engine query log and DBpedia [1] and estimate their typicality using frequencies of class/attribute and instance/attribute occurrences. The extraction process is completely label-based. For the merging of attributes, they use synonyms derived from Wikipedia.

Several systems have been proposed to extend a user-specified query table with content from a corpus of Web tables [2, 16, 10]. For the task of finding new attributes, the user can specify a keyword query which describes the new attribute, so no ranking is required. Alternatively, the InfoGather system [16] and the Mannheim SearchJoin Engine [10] can generate additional attributes based on a schema matching, but both systems do not rank the resulting attributes based on a relevance score.

3 Approach

Our goal is to design an ontology augmentation solution to find new attributes for an ontology using an external source of structured data, such as a corpus of web tables. The general idea followed by existing approaches is to count attribute occurrences in the table corpus and use them to estimate probabilities for encountering these attributes. Based on these probabilities, several different metrics can be defined to assess the value of adding an attribute to the ontology (see Section 3.3). These metrics measure how likely a new attribute is to co-occur with existing attributes (in the ontology) or how consistent the resulting schema would be if the new attribute is added to the ontology.

Existing methods often consider the use-case of extending a user-provided data source in an ad-hoc setting. In the case of extending an ontology, however, a variety of matching methods can be used to align the schema of the Web tables with the ontology. We propose to incorporate the mapping created by such methods by calculating all co-occurrence frequencies based on the mapping.

The relevance of new attributes is measured based on how frequently they co-occur with known attributes. Using exact string matching, these frequencies can be obtained from a corpus of web tables by counting, as shown by the approaches using the AcsDb [3]. When using fuzzy matching methods, however, the attributes must first be mapped among each other and then be partitioned according to their similarity values. This results in attribute clusters whose frequency can be determined by adding up the frequencies of all attributes in the cluster.

3.1 Identifying Equal Attributes

We compare several different approaches of defining attribute similarity, which will be introduced in the following.

Equality of Known Attributes. For the attributes that already exist in the ontology, we create a mapping from the web tables to the ontology. For the results in this paper, we use T2K Match [12] to map Web Tables to DBpedia ontology. This mapping defines which columns in the web tables correspond to which property in the ontology. By transitivity, all attributes which correspond to the same property are equal.

Equality of Unknown Attributes. Based on the mapping produced by T2K Match, we can group the web tables by their class in the knowledge base (blocking step) and then match all un-mapped attributes among each other. For attributes which do not exist in the ontology, we compare the following schema matching approaches:

Label-based Matching. Using the column headers of web tables as features, we evaluate using exact column header equality to find matching columns. We refer to this approach as “Exact” in our experiments. We further evaluate “String Similarity”, which calculates the similarity of the column headers using the Generalised Jaccard Similarity with Edit Distance as inner similarity function.

Instance-based Equality. We further evaluate similarities which are created by the instance-based schema matcher of the Helix System [6]. We refer to the configuration using cosine similarity as “Helix Cosine” and to the configuration using containment similarity as “Helix Containment”.

Key/Value-based Equality. The Key/Value-based equality “Key/Value Matching” compares only those values of two columns, which are mapped to the same instance in the ontology. This means, two columns are equivalent only if they contain similar values for the same instances. To obtain the similarity values, we use the value-based matching component of T2K Match.

3.2 Similarity Graph Partitioning

After the calculation of the similarity values, we must decide which set of columns refers to the same attribute. For attributes that already exist in the ontology, all columns with a similarity value which is above a threshold are considered to be equal to the existing attribute. However, for attributes which do not exist in the ontology, there is no such central attribute. We hence evaluate different partitioning strategies [8] for the graph that is defined by the similarities among the columns of the web tables.

Connected Components. We calculate the connected components on the similarity graph. Each resulting component is a cluster.

Center. The Center algorithm uses the list of similarities sorted in descending order to create star-shaped clusters. The first time a node is encountered in the sorted list, it becomes the center of a cluster. Any other node appearing in a similarity pair with this node is then assigned to the cluster having the former node as center.

MergeCenter. The MergeCenter algorithm is similar to the Center algorithm, but has one extension. This extension is that if a node is similar to the centers of two different clusters, these clusters are merged together.

3.3 Attribute Ranking

After defining attribute equality, we can now specify how the relevance of new attributes is determined. All compared ranking methods are defined based on attribute cooccurrence probabilities, which we define according to Cafarella et al. [3].

Let a schema $s \in S$ be a set of attributes and S be the set of all schemata. A table has this schema if its columns correspond to the attributes (based on the schema mapping), regardless of their order and column header. Let $freq(s)$ be the number of tables with schema s in the corpus and $schema\_freq(a)$ be the number of tables that contain attribute a:

$$schema\_freq(a) = \sum_{\{s \mid s \in S \wedge a \in s\}} freq(s) \tag{1}$$

Then the probability of encountering a in any table in the corpus is

$$p(a) = \frac{schema\_freq(a)}{\sum_{s \in S} freq(s)} \tag{2}$$

The number of tables that contain two attributes a_1, a_2 is defined analogously as $schema\_freq(a_1, a_2)$. The conditional probability of seeing attribute a_1 given a_2 is

$$p(a_1|a_2) = \frac{schema\_freq(a_1, a_2)}{schema\_freq(a_2)} \tag{3}$$

And the joint probability is

$$p(a_1, a_2) = \frac{schema\_freq(a_1, a_2)}{\sum_{s \in S} freq(s)} \tag{4}$$

Attribute Ranking Methods. We now define the methods that are used to calculate a score for each attribute, which is then used to rank all unknown attributes. A higher score indicates a higher relevance of the attribute for the schema extension task.

Conditional Probability based on Class. Given the class C in the ontology, how likely is it to encounter the attribute a [9]. If each schema is mapped to a class C and $schema\_freq(a, C)$ is the number of tables mapped to C that contain a, we can define the conditional probability of encountering an attribute based on the class as in Equation 5, where S_C is the schema of class C. This measure only considers the class mapping of the web tables, irrespective of the presence of known attributes in the same web table.

$$p(a|C) = \frac{schema\_freq(a, C)}{\sum_{a_2 \in S_C} schema\_freq(a_2, C)} \tag{5}$$

Schema Consistency. This measure reflects the likelihood of seeing a new attribute together with the existing attributes [4]. It is based on the conditional probability derived from the cooccurrence statistics. This measure considers all known attributes which co-occur with the new attribute a, i.e., the more known attributes co-occur, the higher the score.

$$SchemaConsistency(a, s) = \frac{1}{|s|} \cdot \sum_{a_2 \in s} p(a|a_2) \tag{6}$$

Schema Coherency. Based on Point-wise Mutual Information (PMI), schema coherency is the average of the PMI scores of all possible attribute combinations [3]. The PMI score of two attributes is positive if the attributes are correlated, zero if they are independent, and negative if they are negatively correlated.

$$SchemaCoherency(a, s) = \frac{1}{|s|} \cdot \sum_{a_1 \in s} npmi(a_1, a) \tag{7}$$

$$npmi(a_1, a_2) = -\frac{1}{\log p(a_1, a_2)} \cdot \log \frac{p(a_1, a_2)}{p(a_1) \cdot p(a_2)} \tag{8}$$

4 Experiments

4.1 Experiments on T2D Gold Standard

Our first set of experiments are performed on the T2D Gold Standard [12], which was originally developed to evaluate systems for the web table to knowledge base matching task (using DBpedia as the knowledge base).

Identifying equal Attributes We evaluate the different matching and partitioning approaches introduced in Section 3.1. The gold standard contains mappings from the web table columns to properties in the ontology. As we are interested in finding partitions of columns which represent the same attribute, we create one partition for each property in the ontology, which contains all columns which are mapped to this property. We then apply the different methods to all columns of the web tables in the gold standard and measure the degree to which we can reconstruct these partitions.

Figure 1 shows a comparison of the different partitioning approaches. The x-axis depicts the similarity threshold and the y-axis shows the resulting F1-score. We can see that the best performance is achieve with rather low thresholds and the Center algorithm.

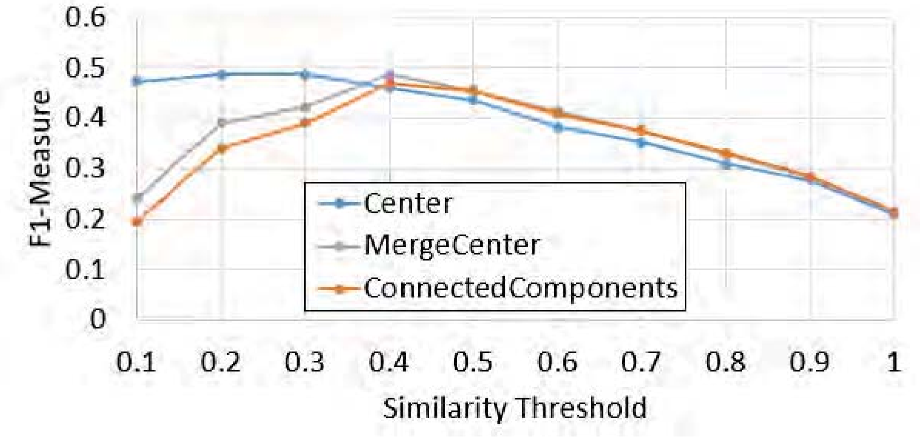


Fig. 1. Evaluation of similarity graph partitioning methods.

Figure 2 shows the quality of the Clusterings using different matching approaches. Again, the x-axis depicts the similarity threshold and the y-axis shows the resulting F1-score. We see that the label-based matching with string similarity outperforms the instance-based approaches. The reason for the good performance of the label-based approach is that the web tables are grouped by the class in the ontology to which they are mapped, and hence column headers are in most cases not ambiguous. The rather bad performance of the instance-based approaches is explained by the fact that web tables usually only have very few rows and there might not be enough overlap among the columns from different tables.

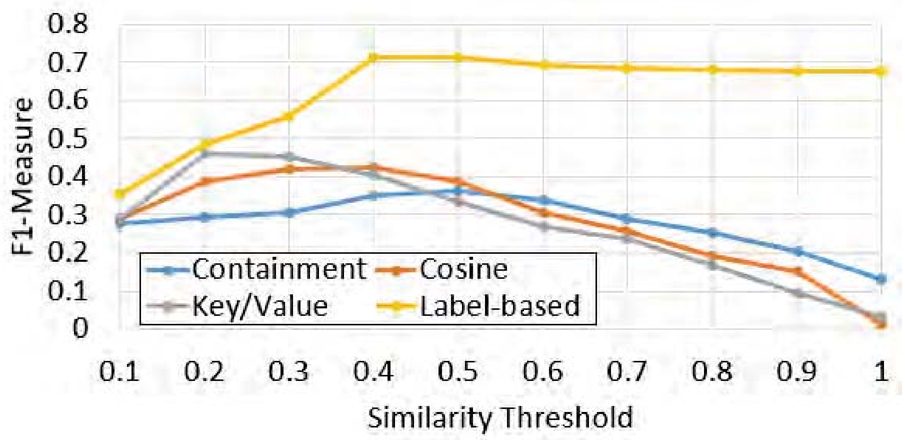


Fig. 2. Evaluation of matching methods.

Combining the instance-based and label-based approach into a hybrid matcher did not significantly improve the performance compared to the label-based approach. Closer inspection of the results showed that this is due to the used gold standard, which contains mostly tables with columns labels of high quality.

Attribute Ranking For a subset of the T2D tables, those mapped to the country class, we manually label all columns with either "useful" or "not useful". In total, this subset contains 207 columns, of which 86 are annotated as "useful". We then evaluate the performance of the different ranking methods. Figure 3 shows the precision@K and recall@K achieved by the different ranking approaches. In addition to the ranking methods described in Section 3.3, we further evaluate each of the ranking methods in a variant that is weighted by PageRank. The intuition is that web pages with a high PageRank likely contain useful content and hence the web tables on these pages also contain relevant attributes. The used PageRank values are obtained from the publicly available Common Crawl WWW Ranking.[4] For each partition of columns, we use the maximum PageRank of all source web pages and multiply it with the score that was calculated by the ranking method. Among the different ranking methods, schema consistency performs best, followed by schema coherency. The variations with PageRank

---

[4] http://wwwranking.webdatacommons.org/

perform worst, which might be caused by the rather small number of web sites in the gold standard.

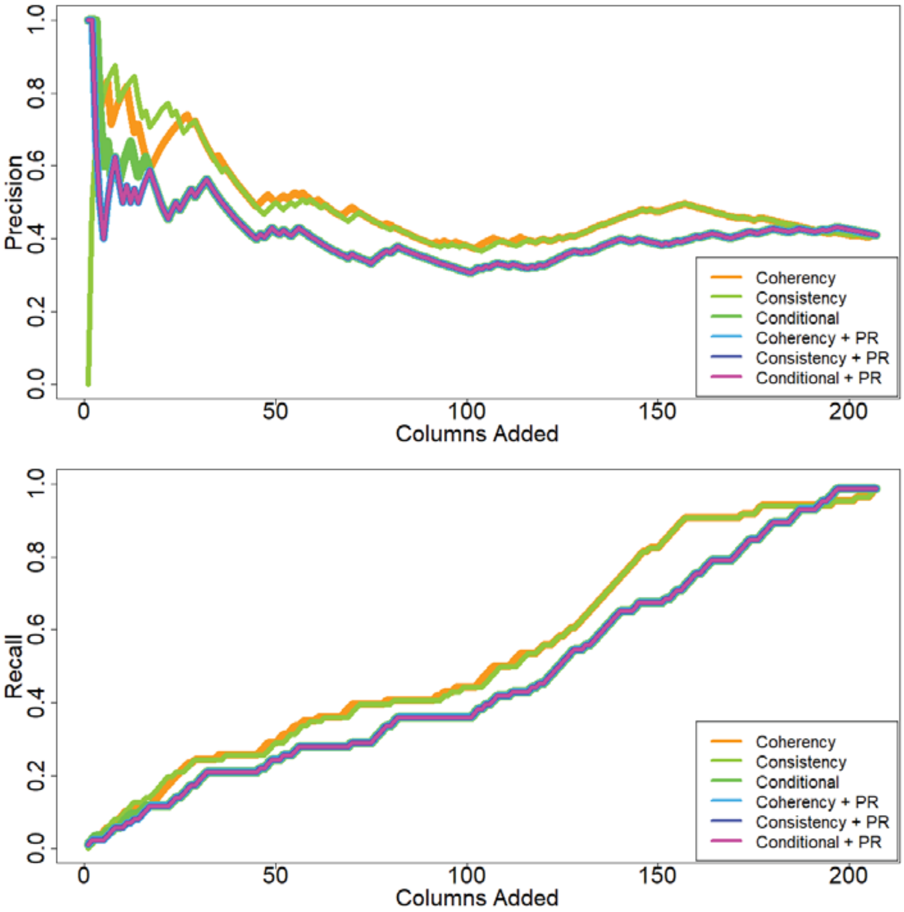


Fig. 3. Precision@K and Recall@K achieved by the different ranking methods using the Key/Value matcher.

Remove one Attribute Experiment The assessment of the usefulness of an attribute can be subjective. Hence we design another experiment, where we remove one existing attribute from the ontology for several classes. As this attribute was already existing, we can objectively say that it is useful. We then measure the quality of the first cluster that resembles this attribute and also the rank at which we can find it in the output. We use the following classes and attributes in this experiment: *Company* (*industry*), *Country* (*population*), *Film* (*year*), *Mountain* (*height*), *Plant* (*family*), *VideoGame* (*genre*). The left chart in Figure 4 shows the average rank of the first attribute cluster which matches the removed attribute over all ranking methods by matcher. The bar "No Matching" shows the result of neither using correspondences to the ontology nor any of the matching approaches, i.e., attributes are equal only if their column headers match exactly. The results without prior mapping knowledge show the importance of matching attributes before calculating the ranking functions. Without mapping knowledge, attribute frequencies are under-estimated, and the respective attribute is ranked too low. The right chart in Figure 4 shows the average rank over all matching methods by ranking method. Again, the schema consistency ranking performs best and the variations including PageRank consistently perform worse.

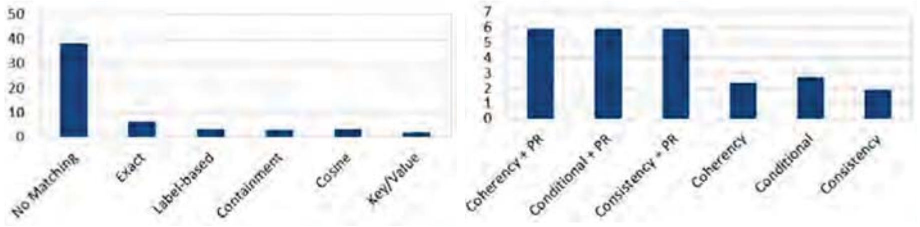


Fig. 4. Rank of the first cluster matching the removed attribute. Left: by matcher. Right: by ranking method.

4.2 Experiments on WDC Table Corpus

We now repeat our experiments on the WDC Web Tables Corpus 2012[5], which contains 147 million relational web tables. To give an overall impression of the full corpus, Figure 5 shows the number of new columns and clusters that we can generate for selected classes. These numbers show the large amount of potentially new attributes that can be found in the corpus.

Attribute Ranking As we have no gold standard for the full corpus, we manually annotate the top 15 ranked clusters for each ranking method for several classes with either "useful" or "not useful". Figure 5 shows the performance of each method averaged over all classes in terms of precision@15. The results show again that the schema coherency and consistency measures outperform the conditional measure. This indicates that attribute co-occurrence is a stronger signal than pure frequency of attributes, even if conditioned with a class from the ontology.

---

[5] http://webdatacommons.org/webtables/index.html

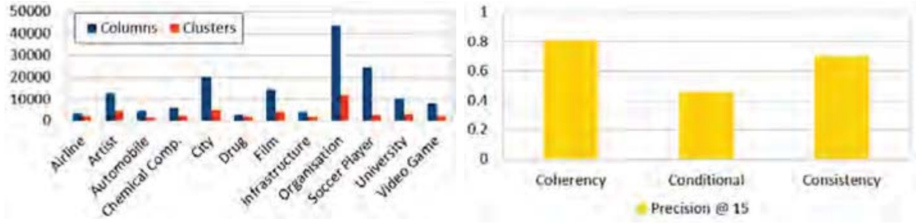


Fig. 5. Left: Number of attributes and clusters, that do not exist in the ontology. Right: Manual evaluation of the usefulness of new attributes.

Remove one Attribute Experiment Again, to have a more objective view on the results, we remove one attribute from DBpedia as before and find the top-ranked attribute cluster which matches the removed attribute. The left chart in Figure 6 shows the rank of these clusters by matcher and the right chart by ranking method. Concerning the matching approach, we now find that the label-based and key/value-based methods achieve comparable results. The difference here to the experiment on the gold standard is that we take into account a much larger number of tables and hence have more variety and a more realistic sample of the data quality. If we compare both of the matching approaches to a baseline approach ("No Matching"), which does not use the prior knowledge of the mappings to the ontology, we can again see that the ranking results are worse. Looking at the different ranking methods, we see a result that differs from the previous results. The Conditional Probability ranking now performs best. A possible explanation is that the attributes that we removed are quite common. Hence, many tables have such attributes and the ranking by frequency is sufficient. Another interesting fact is that now the PageRank makes a difference. Although it is still worse than without, we can presume that a reasonable evaluation of a ranking method incorporating the PageRank requires the use of a large corpus.

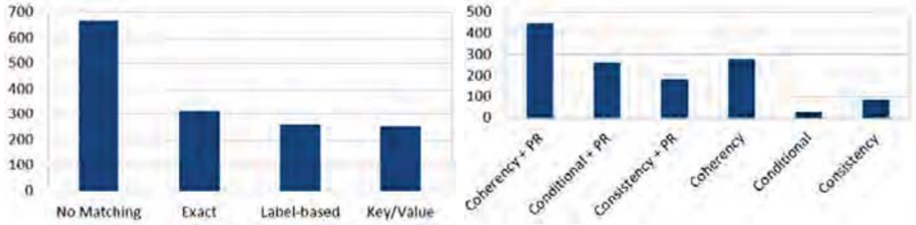


Fig. 6. Left: Rank of the first cluster matching the removed attribute by matcher. Right: Rank of the first cluster matching the removed attribute.

5 Conclusion & Future Work

In summary, the results of our experiments show that:

- It is feasible to use a large corpus of structured data from the Web to augment an ontology. In particular, we are able to augment a general-domain ontology such as DBpedia with millions of Web Tables extracted from the Web. We manually verified that a number of attributes ranked highly by our algorithms were strong candidates for augmenting the DBpedia ontology, and such augmentations would enable new applications of the ontology.
- Our results comparing different algorithms were mixed and without a clear winner across all the experiments. The size of the gold standard and the classes chosen for manual verification clearly affected the relative performance of the algorithms. This calls for larger benchmarks, more comprehensive evaluation, and hybrid/ensemble methods that effectively take advantage of the benefits of each of the algorithms.

Future work also includes: 1) extending our framework to include more advanced matching techniques particularly from recent work in ontology matching 2) evaluation on other sources of structured data (e.g., open data portals such as data.gov), and other ontologies 3) Extending the augmentation to relations and classes of the ontology 4) using the same quality metrics for ontology augmentation from textual and semi-structured sources and an evaluation of how well structured data on the Web can contribute to building and augmenting an ontology, comparing with the textual and semi-structured sources.

References

1. Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. *Dbpedia: A nucleus for a web of open data*. Springer, 2007.
2. Michael J Cafarella, Alon Halevy, and Nodira Khoussainova. Data integration for the relational web. *Proceedings of the VLDB Endowment*, 2(1):1090–1101, 2009.
3. Michael J Cafarella, Alon Halevy, Daisy Zhe Wang, Eugene Wu, and Yang Zhang. Webtables: exploring the power of tables on the web. *Proceedings of the VLDB Endowment*, 1(1):538–549, 2008.
4. Anish Das Sarma, Lujun Fang, Nitin Gupta, Alon Halevy, Hongrae Lee, Fei Wu, Reynold Xin, and Cong Yu. Finding related tables. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, pages 817–828. ACM, 2012.
5. Xin Dong, Evgeniy Gabrilovich, Geremy Heitz, Wilko Horn, Ni Lao, Kevin Murphy, Thomas Strohmann, Shaohua Sun, and Wei Zhang. Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 601–610. ACM, 2014.
6. Jason Ellis, Achille Fokoue, Oktie Hassanzadeh, Anastasios Kementsietsidis, Kavitha Srinivas, and Michael J Ward. Exploring big data with helix: Finding needles in a big haystack. *ACM SIGMOD Record*, 43(4):43–54, 2015.

7. Rahul Gupta, Alon Halevy, Xuezhi Wang, Steven Euijong Whang, and Fei Wu. Biperpedia: An ontology for search applications. *Proceedings of the VLDB Endowment*, 7(7):505–516, 2014.
8. Oktie Hassanzadeh, Fei Chiang, Hyun Chul Lee, and Renée J Miller. Framework for evaluating clustering algorithms in duplicate detection. *Proceedings of the VLDB Endowment*, 2(1):1282–1293, 2009.
9. Taesung Lee, Zhongyuan Wang, Haixun Wang, and Seung-won Hwang. Attribute extraction and scoring: A probabilistic approach. In *Data Engineering (ICDE), 2013 IEEE 29th International Conference on*, pages 194–205. IEEE, 2013.
10. Oliver Lehmberg, Dominique Ritze, Petar Ristoski, Robert Meusel, Heiko Paulheim, and Christian Bizer. The mannheim search join engine. *Web Semantics: Science, Services and Agents on the World Wide Web*, 2015.
11. Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. A review of relational machine learning for knowledge graphs. *Proceedings of the IEEE*, 104(1):11–33, 2016.
12. Dominique Ritze, Oliver Lehmberg, and Christian Bizer. Matching html tables to dbpedia. In *Proceedings of the 5th International Conference on Web Intelligence, Mining and Semantics*, page 10. ACM, 2015.
13. Dominique Ritze, Oliver Lehmberg, Yaser Oulabi, and Christian Bizer. Profiling the potential of web tables for augmenting cross-domain knowledge bases. In *Proceedings of the 25th international conference on world wide web*, pages 251–261. International World Wide Web Conferences Steering Committee, 2016.
14. Petros Venetis, Alon Halevy, Jayant Madhavan, Marius Paşca, Warren Shen, Fei Wu, Gengxin Miao, and Chung Wu. Recovering semantics of tables on the web. *Proceedings of the VLDB Endowment*, 4(9):528–538, 2011.
15. Wentao Wu, Hongsong Li, Haixun Wang, and Kenny Q Zhu. Probase: A probabilistic taxonomy for text understanding. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, pages 481–492. ACM, 2012.
16. Mohamed Yakout, Kris Ganjam, Kaushik Chakrabarti, and Surajit Chaudhuri. Infogather: entity augmentation and attribute discovery by holistic matching with web tables. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, pages 97–108. ACM, 2012.

Introducing the HOBBIT platform into the Ontology Alignment Evaluation Campaign

Ernesto Jiménez-Ruiz[1,2], Tzanina Saveta[3], Ondřej Zamazal[4], Sven Hertling[5],
Michael Röder[6], Irini Fundulaki[3], Axel-Cyrille Ngonga Ngomo[6],
Mohamed Ahmed Sherif[6], Amina Annane[7,8], Zohra Bellahsene[8], Sadok Ben Yahia[9],
Gayo Diallo[10], Daniel Faria[11], Marouen Kachroudi[9], Abderrahmane Khiat[12],
Patrick Lambrix[13], Huanyu Li[13], Maximilian Mackeprang[12], Majid Mohammadi[14],
Maciej Rybinski[15], Booma Sowkarthiga Balasubramani[16] and Cassia Trojahn[17]

[1] The Alan Turing Institute, London, United Kingdom
[2] Department of Informatics, University of Oslo, Norway
[3] Institute of Computer Science - FORTH, Greece
[4] University of Economics, Prague, Czech Republic
[5] Data and Web Science Group, University of Mannheim, Germany
[6] Paderborn University, Data Science Group, Pohlweg 51, D-33098 Paderborn, Germany
[7] Ecole nationale Superieure d'Informatique, Alger, Algerie
[8] LIRMM, Universit de Montpellier, CNRS, Montpellier, France
[9] Université de Tunis El Manar, LIPAH-LR 1 ES14, 2092, Tunis, Tunisie
[10] BPH Center - INSERM U1219, Team ERIAS & LaBRI UMR5800, University of Bordeaux
[11] Instituto Gulbenkian de Ciência, Portugal
[12] Human-Centered Computing, Freie Universitat Berlin, 4195 Berlin, Germany
[13] Linköping University & Swedish e-Science Research Centre, Linköping, Sweden
[14] Delft University of Technology, The Netherlands
[15] Dpto. Lenguajes y Ciencias de la Computación, University of Malaga, Spain
[16] ADVIS Lab, Department of Computer Science, University of Illinois at Chicago, USA
[17] Institut de Recherche en Informatique de Toulouse, France

Abstract. This paper describes the Ontology Alignment Evaluation Initiative 2017.5 pre-campaign. Like in 2012, when we transitioned the evaluation to the SEALS platform, we have also conducted a pre-campaign to assess the feasibility of moving to the HOBBIT platform. We report the experiences of this pre-campaign and discuss the future steps for the OAEI.

1 Introduction

The Ontology Alignment Evaluation Initiative[1] (OAEI) is a coordinated international initiative which organizes the evaluation of ontology matching systems [1,2]. The main goal of the OAEI is to compare systems and algorithms openly and on the same basis to allow anyone to draw conclusions about the best matching strategies. Furthermore, our ambition is to help tool developers to improve their systems through such evaluations.

The initiative started in 2004, and from 2006 until the present, the OAEI campaigns were held at the Ontology Matching workshop, collocated with the ISWC conference. Since 2011, we have been using an environment for automatically processing evaluations which was developed within the SEALS (Semantic Evaluation At Large Scale)

---

[1] `http://oaei.ontologymatching.org`

project[2]. SEALS provided a software infrastructure for automatically executing evaluations and evaluation campaigns for typical semantic web tools, including ontology matching. In the OAEI 2017, a novel evaluation environment called HOBBIT was adopted for the novel HOBBIT Link Discovery track. In that OAEI campaign, all systems were executed under the SEALS client in all other tracks.

The good experience of the 2017 Link Discovery (*e.g.*, novel platform, online evaluation, automatic generation of result tables, attraction of link discovery developers, customization of the matching requirements of a benchmark task) track motivated the interest in assessing the possibility of transitioning the whole OAEI evaluation to HOBBIT. To that end, we decided to set-up an OAEI pre-campaign, as happened in the OAEI 2011.5 when the OAEI moved to SEALS,[3] to evaluate potential risks and challenges. The nature of the link discovery tracks is different from the (traditional) OAEI tracks and we foresaw sources of uncertainty with respect to: *(i)* the use of a new evaluation environment, *(ii)* the adaptation of tracks with multiple tasks (like *multifarm*), *(iii)* the introduction of Docker to organisers and participants, *(iv)* the inclusion of interactivity capabilities, and *(v)* the storage of results. The objective of the Ontology Alignment Evaluation Initiative 2017.5 pre-campaign was, therefore, to evaluate the feasibility of moving some (traditional) OAEI tracks to the HOBBIT platform. In this paper, we report the experiences of this pre-campaign and future steps of the OAEI.

The remainder of the paper is organised as follows. Section 2 introduces the HOBBIT platform. In Section 3, we present the overall evaluation methodology that has been used. Section 4 describes the evaluation data sets and Section 5 the participating systems. Section 6 overviews the lessons learned from the campaign; and finally, Section 7 summarizes the conclusions of this experience and discusses future plans for the OAEI.

2 HOBBIT platform

The HOBBIT platform is a generic, modular and distributed platform for Big Linked Data systems. It was designed to enable Big Data practitioners and Linked Data users to benchmark all steps of the data lifecycle at scale, i.e., with all necessary contemplations of volume, velocity, value and veracity necessary to benchmark real applications. Some of its most important features within the context of link discovery include the support of *(i)* benchmarks that focus on the evaluation of the quality of a system using single consecutive requests as well as *(ii)* benchmarks aiming at evaluating the efficiency of Big Linked Data solutions, e.g., by generating distributed parallel requests leading to a high workload. The HOBBIT project[4] designed and develops the HOBBIT platform with the aim of providing an open-source, extensible, FAIR[5] and scalable evaluation platform (in a fashion akin to GERBIL [3]) along with corresponding benchmarks and mimicking algorithms for real data sources of industrial scale. The platform being open-source means that it can be downloaded and installed locally for tests. The online instance of the platform allows *(i)* running public challenges and *(ii)* making sure that even people without the required infrastructure are able to run the benchmarks they are interested in. The platform, as well as the benchmarks that are designed and implemented in HOBBIT

---

[2] `http://www.seals-project.eu`

[3] `http://oaei.ontologymatching.org/2011.5/`

[4] `http://project-hobbit.eu`

[5] Findable, Accessible, Interoperable and Reusable

are modelled as actors with which the platform interacts. The use cases relevant for end users supported by the platform include:

- *Benchmarking a System:* the user can select a benchmark to test a system. The platform loads appropriate configuration parameters for the benchmark, as well as the list of available systems for this benchmark. The user configures the benchmark and selects one of the available systems to benchmark.
- *Showing and Comparing Benchmark Results*: the user can view the results of a single benchmark run or select multiple, e.g., to compare several systems that have been evaluated with the same benchmark.
- *Adding a System:* the user adds the system that needs to be benchmarked in the platform by providing a docker image of the system and a system adapter which serves as a proxy between the benchmark and the system.

Figure 1 shows the layout of the HOBBIT platform components and how the different parts interact. The platform can be separated into two parts. The first part comprises platform components that are always running (right hand side of Figure 1). The second part contains all components that belong to a certain experiment (left hand side of Figure 1), i.e., the benchmark components as well as the benchmarked system.

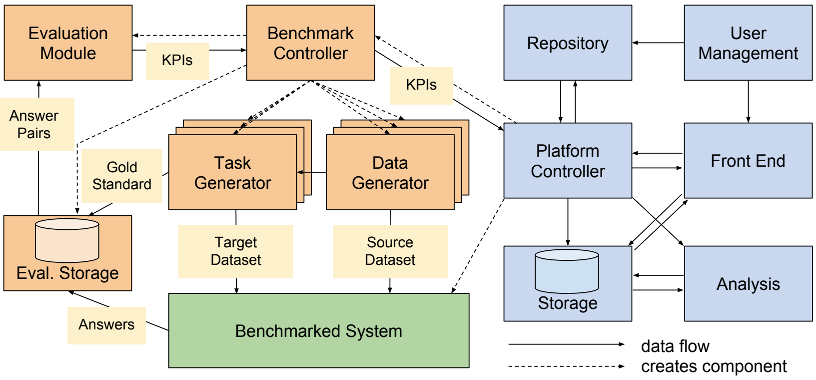


Fig. 1. Interaction of the components of HOBBIT Platform

The *Platform Controller* makes sure that the benchmark chosen by the user can be started and ensures that all nodes of the cluster are available. It communicates with the system to be benchmarked, ensures that it is working properly and generates the benchmark controller that is responsible for producing the data and task generators as well as the evaluation storage. The *Data Generator* produces the source dataset that is sent to the *Benchmarked System*, and the target dataset as well as the Gold Standard which are sent to the *Task Generator*. The *Task Generator* sends the target dataset to the *Benchmarked System* and forwards the Gold Standard to the *Evaluation Storage*. When the system finishes its task, it sends the answers to the *Evaluation Storage*. The *Evaluation Module* receives the system and the Gold Standard answers and returns the Key Performance Indicators (KPIs) for the experiment.

3 Methodology

The OAEI campaigns are typically divided into three phases: *(i)* preparation phase (datasets are prepared), *(ii)* execution phase (systems are tuned), and *(iii)* evaluation phase (systems are evaluated). In this OAEI pre-campaign, we focused on the preparation and execution phases given the time constraints and the challenges encountered during the migration to the HOBBIT platform.

3.1 Preparation phase

The preparation phase for the OAEI 2017.5 pre-campaign, unlike recent OAEI campaigns, was more demanding as the OAEI track organisers were required to migrate the SEALS datasets following the novel HOBBIT specifications. We provided the benchmark definitions for the *(i)* Largebio[6] and *(ii)* Link discovery[7] tracks to make the transition smoother. These (reference) datasets were made available by the end of January 2018. Next, we provide a brief summary of the main components of a HOBBIT benchmark.

HOBBIT *benchmark definition.* The HOBBIT workflow and format of benchmarks is generic as the platform was designed to accommodate benchmarks across the whole of the Linked Data lifecycle. This flexibility adds some complexity with respect to the SEALS benchmark generation. Note that, since in the OAEI multi-tasks benchmarks the source dataset may change, we have slightly modified the general HOBBIT workflow depicted in Figure 1. In the OAEI workflow, the TaskGenerator deals with both the source and target datasets to generate a Task. A benchmark is composed by the following classes:

BenchmarkController is the main class of the benchmark where the general benchmark execution workflow is specified.

DataGenerator generates the benchmark datasets (*e.g.*, input ontologies and alignments) and prepares the datasets for the TaskGenerator. For multiple-task benchmarks it also deals with the preparation of queue names to be sent to the system.

Task includes the information of the source and target datasets and the expected results together with some other parameters like which type of entity should be matched (*e.g.*, only classes).

TaskGenerator deals with the generation of the task(s) and sends the task(s) to the system and the *EvaluationModule*.

EvaluationModule compares the expected results (*e.g.*, reference alignment) provided by the *TaskGenerator* and the computed results by a system, and generates the KPIs.

Each benchmark is also associated to a metadata file[8] where the docker images of the benchmark are referenced, the KPIs defined, and the name of the benchmark's API specified (*e.g.*, *bench:LargebioAPI*).

---

[6] `https://gitlab.com/ernesto.jimenez.ruiz/largebio`

[7] `https://github.com/hobbit-project/SpatialBenchmark/`

[8] Metadata for largebio: `https://git.project-hobbit.eu/ernestoj/largebio`

Table 1. OAEI 2017.5 Benchmarks: HOBBIT APIs and KPIs. Each benchmark has its own API as they may define different input parameters. Only systems compliant with (*i.e.*, implementing) the API will be evaluated under a given benchmark.

Track	**API**	**KPIs**
Conference	bench:ConferenceAPI	Precision, Recall, F-measure
Anatomy	bench:AnatomyAPI	Precision, Recall, F-measure, Recall+
Largebio	bench:LargebioAPI	Precision, Recall, F-measure
Spimbench	bench:spimbenchAPI	Precision, Recall, F-measure
Link discovery	bench:LinkingAPI	Precision, Recall, F-measure

OAEI 2017.5 tracks. The preparation phase was complete in early March 2018 and led to four novel tracks running under the HOBBIT platform: *conference*, *anatomy*, *largebio*, and *instance matching - spimbench*. Note that the *link discovery* track was already running under HOBBIT in the OAEI 2017 campaign. The benchmarks are (briefly) described in Section 4.

3.2 Execution phase

The execution phase also brought the new challenge to developers of implementing a system compliant with the HOBBIT specifications. We provided the following sources of instruction to support system developers with the integration with HOBBIT: *(i)* General HOBBIT instructions,[9] *(ii)* LogMap's example implementing the interfaces for the *conference*, *anatomy*, *largebio* and *spimbench* tracks[10], and *(iii)* the Maven framework to facilitate the wrapping of systems.[11]

HOBBIT *system definition.* The interface of a system is defined via the *SystemAdapter* class (e.g., LogMapSystemAdapter). This class receives the dataset definition from the *DataGenerator* of a benchmark (e.g. set of tasks and matching requirements) and the individual tasks (source and target datasets) from the *TaskGenerator* of a benchmark. The results (e.g., a file containing the mappings in RDF Alignment format) are sent to the benchmark's *EvaluationModule*. The system adapter class communicates to the benchmark classes in a special way since it is submitted to the HOBBIT platform as a docker image. Each system is also associated to a metadata file,[12] which explicitly mentions the APIs the system implements (e.g. hobbit:implementsAPI bench:LargebioAPI). This enables the automation of the evaluation of the OAEI benchmarks.

OAEI 2017.5 participation. Ten systems were registered to participate in the OAEI 2017.5 campaign in March 2018. Only eight of them reported results or experiences during April and May: *OntoIdea*, *LogMap*, *SANOM*, *DisMatch*, *KEPLER*, *YAM-BIO*, *AML* and *RADON*. The participating system and proof-of-concept results are (briefly) presented in Section 5.

---

[9] `https://project-hobbit.eu/challenges/oaei2017-5/oaei2017-5-tasks/`

[10] LogMap [4]: `https://gitlab.com/ernesto.jimenez.ruiz/logmap-hobbit`

[11] Maven framework: `https://github.com/sven-h/ontMatchingHobbit`

[12] LogMap's metadata: `https://git.project-hobbit.eu/ernestoj/logmapsystem`

4 Benchmarks

The OAEI 2017.5 pre-campaign included five tracks: *conference*, *anatomy*, *largebio*, *instance matching - spimbench*, and *link discovery*. Table 1 provides a summary of the benchmarks. This pre-campaign did not include the *multifarm* and the *interactive* tracks. In the case of the *multifarm* track the main restriction was to move thousands of matching tasks to a new environment. While for the *interactive* tracks the main limitation was technological as the inclusion of an "oracle" requires significant modifications on the HOBBIT pipeline. Next we briefly describe the datasets of the OAEI 2017.5 benchmarks.

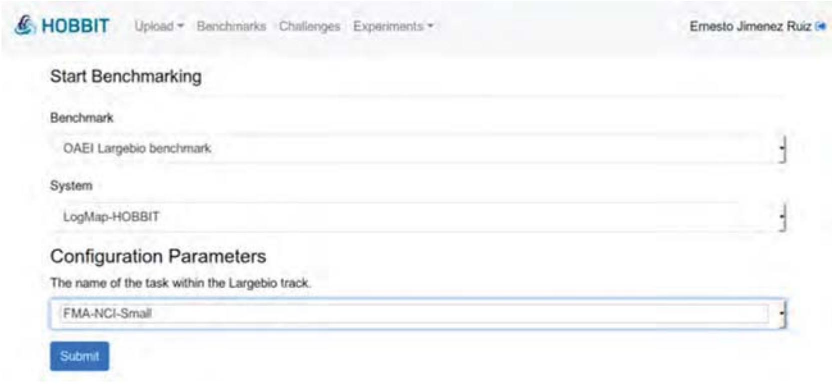
Anatomy track. This track consists of finding an alignment between the Adult Mouse Anatomy ontology (AMA) and a part of the National Cancer Institute (NCI) Thesaurus (NCI-A). This data set has been used since 2007 with some improvements over the years [5]. The AMA ontology contains 2,744, while the NCI-A contains 3,304 concepts describing the human anatomy. Systems participating in the anatomy track are evaluated in terms of runtime, precision, recall and F-measure. In addition, the anatomy track measures the systems' ability to find non-trivial correspondences (recall+) and checks whether the systems generate coherent alignments.

Conference track. This track consists of 21 test cases with ontologies from the domain of organising conferences. The conference track has been used since 2006 and it was gradually improved [6]. The advantage of the conference domain is the fact that it is generally understandable. The ontologies were developed independently and based on different resources, thus they capture the issues in organising conferences from different points of view and using different nomenclature. Finally, ontologies within this track are of small-medium size and relatively rich in OWL 2 axioms.

Largebio track. This track consists of finding alignments between the Foundational Model of Anatomy (FMA), SNOMED CT, and the National Cancer Institute Thesaurus (NCI) [7]. These ontologies are semantically rich and contain tens of thousands of classes. UMLS Metathesaurus has been selected as the basis for the track reference alignments (see [8] for details). UMLS is currently the most comprehensive effort for integrating independently-developed medical thesauri and ontologies, including FMA, SNOMED CT, and NCI. In this track we also put special attention to the number of unsatisfiabilities led by the mappings computed by a participating system.

SPIMBENCH track. The datasets in this strack are produced using SPIMBENCH benchmark generator [9] with the aim to generate descriptions of the same entity where *value-based*, *structure-based* and *semantics-aware* transformations are employed on a source dataset in order to create the target dataset(s). The value-based transformations consider mainly typographical errors and different data formats, the structure-based transformations implement transformations applied on the structure of object and datatype properties and the semantics-aware transformations concern the instance level and take into account schema information. The latter are used to examine if the matching systems take into account RDFS and OWL constructs in order to discover correspondences between instances that can be found only by considering schema information.

Link discovery track. This track is composed of two tasks called: *linking* and *spatial*. The *linking* task measures how well systems can match traces that have been modified using string-based approaches along with addition and deletion of intermediate points.



Start Benchmarking

Benchmark

OAEI Largebio benchmark

System

LogMap-HOBBIT

Configuration Parameters

The name of the task within the Largebio track.

FMA-NCI-Small

Fig. 2. Benchmarking LogMap with the FMA-NCI-SMall *largebio* task.

The original datasets only contain coordinates, thus, we have replaced a number of those points with labels retrieved from Linked Data spatial datasets using the Google Maps[13], Foursquare[14] and Nominatim Openstreetmap[15] APIs to be able apply string-based modifications implemented in LANCE [10]. This task also contains modifications of date and coordinate formats.

The *spatial* task measures how well systems can identify the DE-9IM (Dimensionally Extended nine-Intersection Model) topological relations between *LineStrings* and *Polygons* in two-dimensional spaces. The supported spatial relations are the following: *Equals, Disjoint, Touches, Contains/Within, Covers/CoveredBy, Intersects, Crosses, Overlaps*. The instances are represented in the Well-Known Text (WKT) format. For each relation, a different pair of source and target datasets is given to the participants.

5 Participation and proof-of-concept results

In this section we introduce the systems contributing to the OAEI 2017.5 campaign and provide an overview of how experiments are executed from the HOBBIT public instance.

5.1 HOBBIT experiments

Experiments can be executed via the HOBBIT public instance[16] by following the *Benchmarks* menu. Note that, currently, only registered developers who are the owners of a system which conforms the specification (*i.e.*, API) of one or more benchmarks can run experiments. Figure 2 shows the interface to select a benchmark and evaluate a system implementing its API within the HOBBIT platform.

Every experiment is assigned a unique ID and, once they are finalized, registered users can access its results (see *Experiments* menu). In addition, one can also select several experiments for comparison purposes. For example, Figure 3 shows the results of LogMap for all six tasks of the *largebio* track.

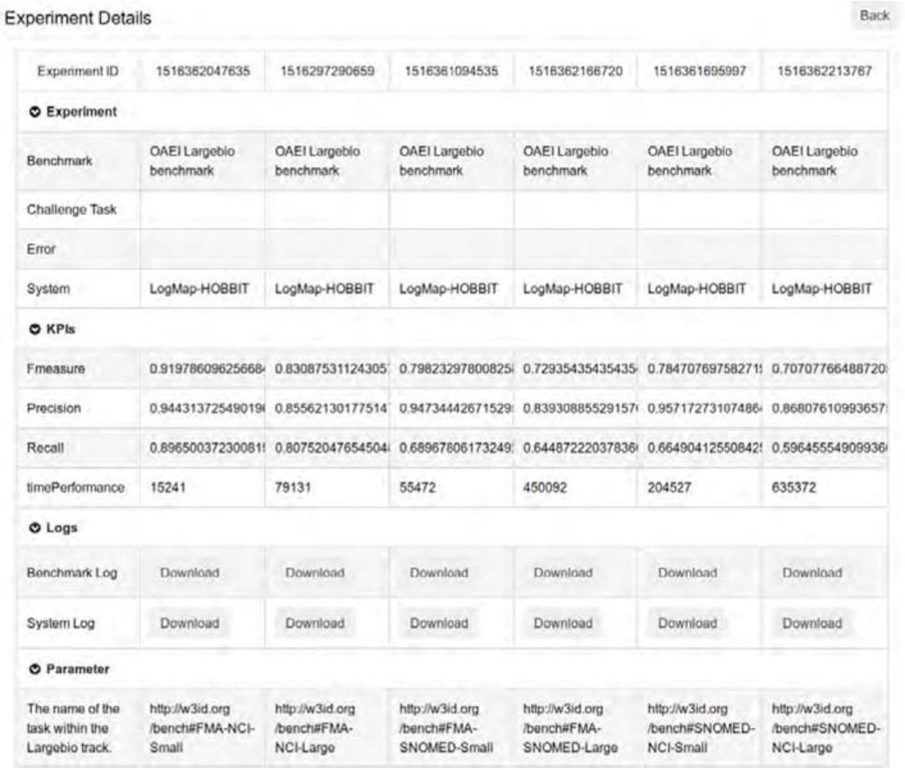
---

[13] `https://developers.google.com/maps/`

[14] `https://developer.foursquare.com/`

[15] `http://nominatim.openstreetmap.org/`

[16] `https://master.project-hobbit.eu/`



Experiment ID	1516382047635	1516297290659	1516361094535	1516362166720	1516361695997	1516362213767
Experiment						
Benchmark	OAEI Largebio benchmark	OAEI Largebio benchmark	OAEI Largebio benchmark	OAEI Largebio benchmark	OAEI Largebio benchmark	OAEI Largebio benchmark
Challenge Task						
Error						
System	LogMap-HOBBIT	LogMap-HOBBIT	LogMap-HOBBIT	LogMap-HOBBIT	LogMap-HOBBIT	LogMap-HOBBIT
KPIs						
Fmeasure	0.91978609625668	0.83087531124305	0.79823297800825	0.72935435435435	0.78470769758271	0.70707766488720
Precision	0.94431372549019	0.85562130177514	0.94734442671529	0.83930885529157	0.95717273107486	0.86807610993657
Recall	0.89650037230081	0.80752047654504	0.68967806173249	0.64487222037836	0.66490412550842	0.59645554909936
timePerformance	15241	79131	55472	450092	204527	635372
Logs						
Benchmark Log	Download	Download	Download	Download	Download	Download
System Log	Download	Download	Download	Download	Download	Download
Parameter						
The name of the task within the Largebio track.	http://w3id.org/bench#FMA-NCI-Small	http://w3id.org/bench#FMA-NCI-Large	http://w3id.org/bench#FMA-SNOMED-Small	http://w3id.org/bench#FMA-SNOMED-Large	http://w3id.org/bench#SNOMED-NCI-Small	http://w3id.org/bench#SNOMED-NCI-Large

Fig. 3. Results of LogMap for all *largebio* tasks.

5.2 System overview

Table 2 lists the participating systems and links to available proof-of-concept results. The developers made a great effort adapting their systems to a new platform providing very useful insights about the feasibility of moving to the HOBBIT platform (see Section 6 for more details). Next, we provide a brief summary of the OAEI 2017.5 systems.

AgreementMakerLight (AML) [11, 12] is an all-purpose ontology alignment system inspired on AgreementMaker [13] and sharing its focus on flexibility and extensibility as main design paradigms. While initially primarily focused on the biomedical domain and on the use of background knowledge, its tool suite and capabilities were gradually extended to cover the full range of ontology matching tasks evaluated under the OAEI.

DisMatch [14] is an experimental ontology matching system built around the idea of leveraging the recent advancements in semantic representations of texts within the context of the ontology alignment problem. The lexical matcher uses semantic similarity calculated from distributional representations of domain-specific words. In the experiments several relatedness measures were tested, based on different text representation methods, including DomESA [15] and Word2Vec's Skip-Gram model [16].

Kepler [17] is an ontology alignment system able to deal with normal and large scale ontologies. Kepler is also able to cope with multilingual ontologies thanks to its translator module. Kepler exploits the expressiveness of the OWL language to detect and

Table 2. Systems participating in the OAEI 2017.5. Link to results requires *guest* log in. DisMatch and KEPLER tested the platform but they did not manage to produce results.

System	New in OAEI/HOBBIT?	Implemented APIs	Link to results
AML	No/No	bench:ConferenceAPI, bench:AnatomyAPI, bench:LargebioAPI, bench:spimbenchAPI, bench:LinkingAPI	`https://goo.gl/ACG3kP`
DisMatch	No/Yes	bench:AnatomyAPI, bench:LargebioAPI,	-
KEPLER	No/Yes	bench:ConferenceAPI, bench:AnatomyAPI, bench:LargebioAPI,	-
LogMap	No/Yes	bench:ConferenceAPI, bench:AnatomyAPI, bench:LargebioAPI, bench:spimbenchAPI	`https://goo.gl/tFDJKB`
OntoIdea	No/No	bench:LinkingAPI	`https://goo.gl/mUjBPK`
RADON	No/No	bench:LinkingAPI	`https://goo.gl/G1nUDY`
SANOM	No/Yes	bench:ConferenceAPI, bench:AnatomyAPI, bench:LargebioAPI,	`https://goo.gl/D8nrJk`
YAM-BIO	No/Yes	bench:AnatomyAPI, bench:LargebioAPI	`https://goo.gl/A496ug`

compute the similarity between ontology entities through six modules: preprocessing, partitioning, translation, indexation, candidate selection and final alignment generation.

LogMap [4] relies on lexical and structural indexes to enhance scalability. It also incorporates approximate reasoning and repair techniques to minimise the number of logical errors in the aligned ontology. LogMap comes with two variants: *LogMap-Bio* [18], which uses BioPortal [19] as a (dynamic) provider of mediating ontologies; and *LogMapLt*, a “lightweight” variant of LogMap that only applies (efficient) string matching techniques.

OntoIdea [20] is an instance matching tool implementing an enhanced version of the STRIM algorithm proposed in previous work [21]. The new version of the *OntoIdea* system identifies not only the “sameAs” relationships between instances, but also the “topological” relationships (*e.g.*, contains, equals, overlaps, covers, etc.) on geo-spatial datasets. The type of relationship is driven by the information associated to the entities (*i.e.*, text or geometry).

RADON [22] is one of the systems of the LIMES framework. It addresses the efficient computation of topological relations on geo-spatial datasets, which belong to the largest sources of Linked Data. The main innovation of the approach is a novel sparse index for geo-spatial resources based on *minimum bounding boxes* (MBB). Based on this index, it is able to discard unnecessary computations for DE-9IM relations. Extensive experiments show that *RADON* scales well and outperforms the state of the art by up to 3 orders of magnitude w.r.t. to its runtime.

SANOM [23] is an ontology alignment system that uses simulated annealing as the principal technique to find correspondences between two given ontologies. The system translates the alignment problem into a state optimization and then applies the simulated annealing to find the optimal alignment of two given ontologies. The optimality of a state is obtained by a complex fitness function which utilizes different lexical and structural similarity metrics.

YAM-BIO is an instance of a generic background knowledge based ontology-matching framework [24] which is publicly available on GitHub[17]. YAM-BIO instance uses YAM++ [25] as matcher and the two biomedical ontologies UBERON and DOID as background knowledge. In the OAEI 2017.5, YAM-BIO adopted a derivation with a specific algorithm that reduces the path number by avoiding to reuse the same background knowledge concept more than once, and the rule-based mapping selection estrategy. YAM-BIO relies on the LogMap-Repair [26] module to eliminate the inconsistent mappings in the generated alignments.

6 Discussion and lessons learned

We collected feedback from eight platform developers pertaining to the transition from SEALS to HOBBIT. A common tenor found in most of the feedback from the systems pertained to the balance between complexity and guarantees. The HOBBIT platform requires *(i)* the systems to be implemented using the Docker stack of technologies, *(ii)* the implementation of a single interface to ensure a set of standardized communication processes and *(iii)* debugging by using log files collected by the platform across the distributed infrastructure it employs for benchmarking.

The use of Docker and associated technologies was largely regarded positively. Whereas the developers unfamiliar with Docker suggested that the supplementary effort necessary to create docker packages was considerable, most developers regarded the use of this technology as a step towards a better integration of tools and more controlled run of benchmarks. To ensure that the development with Docker can be carried efficiently, HOBBIT allows for single Docker files to be ran using different configurations such as to ease the deployment and use. This feature will be made more prominent to ensure that developers make more extensive use thereof.

Participating systems had to implement an API defined by the benchmark to receive the datasets which should be linked and to return the generated results. The participants found a template for this step very helpful and would like to have an even simpler template in the future to reduce the amount of adaptations. Especially when the benchmark API is adapted to support even more complex tasks like the *multifarm* task, a provided template eases the participation. The prepared template could take care of receiving the different ontologies and storing them in single files following the predefined structure of the multifarm task before starting the linking process.[18] Such an extension would enable a backwards compatibility to older solutions which are based on the directory structure. Additionally, a clearer distinction of the necessary and optional steps when implementing the system adapter was rated as helpful.

The online instance of the HOBBIT platform is based on a cluster infrastructure and offers its services to a public community. Since the final evaluation is carried out

---

[17] `https://github.com/AminaANNANE/GenericBKbasedMatcher`

[18] `https://www.irit.fr/recherches/MELODI/multifarm/`

on this instance, the participants were asked to make sure that their systems can be deployed on the platform. The development process in itself can however be eased significantly by testing locally. The HOBBIT platform provides two means for doing so: a software development toolkit (SDK) and a recipe-based deployment infrastructure. The HOBBIT SDK[19] allows to develop and debug a system adapter locally. To this end, the SDK simulates a platform running in a cluster and ensure rapid local development without the overheads (e.g., long waiting times) created by a shared cluster. HOBBIT also supports complete local deployments (instead of simulations) through the recipe-based framework Exoframe[20] for developers who would prefer not using a simulation. Therewith, it allows developers to follow a three-step process: (1) install the HOBBIT SDK or the HOBBIT platform via Exoframe, (2) develop and test your system locally, (3) upload the system image(s) and execute it (them) using the online instance.

7 Conclusions and next OAEI steps

The OAEI 2017.5 pre-campaign was instrumental to understand the strengths of HOBBIT but also the challenges into moving to a new platform. The feedback obtained from system developers has been very valuable for the next steps of the OAEI campaigns and the future development of the HOBBIT platform (*e.g.*, support for more complex tasks, storage of computed alignment). The OAEI 2018 campaign[21] will continue using the HOBBIT platform together with the SEALS infrastructure, with some tracks like *large-bio* providing a dual evaluation mode (*i.e.*, both HOBBIT and SEALS). This way, system developers, organisers and HOBBIT developers will have additional time to guarantee a successful migration to the new evaluation platform. From the infrastructure point of view, the HOBBIT SDK will make the developing and debugging phase under the HOBBIT easier. In addition, we will continue offering the Maven-based framework to facilitate the submission to both HOBBIT and SEALS.

Acknowledgements. EJR was funded by the Centre for Scalable Data Access (SIRIUS), the RCN project BigMed, and the Alan Turing project AIDA. OZ is supported by the CSF grant no. 18-23964S. DF was supported by the ELIXIR-EXCELERATE project (INFRADEV-3-2015). The work of MR is supported by grant TIN2017-86049-R (Ministerio de Economía, Industria y Competitividad, Spain). This work has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 688227 (HOBBIT).

References

1. Euzenat, J., Meilicke, C., Shvaiko, P., Stuckenschmidt, H., Trojahn dos Santos, C.: Ontology alignment evaluation initiative: six years of experience. J. Data Semantics **XV** (2011)
2. Euzenat, J., Shvaiko, P.: Ontology matching. 2nd edn. Springer-Verlag (2013)
3. Röder, M., Usbeck, R., Ngonga Ngomo, A.C.: Gerbil–benchmarking named entity recognition and linking consistently. Semantic Web (Preprint) (2017) 1–21
4. Jiménez-Ruiz, E., Grau, B.C., Zhou, Y., Horrocks, I.: Large-scale interactive ontology matching: Algorithms and implementation. In: Europ. Conf. Artif. Intell. (ECAI). (2012)

---

[19] `https://github.com/hobbit-project/java-sdk`

[20] `https://github.com/hobbit-project/exoframe-recipe-hobbit`

[21] `oaei.ontologymatching.org/2018/`

5. Dragisic, Z., Ivanova, V., Li, H., Lambrix, P.: Experiences from the anatomy track in the ontology alignment evaluation initiative. Journal of biomedical semantics **8**(1) (2017) 56
6. Zamazal, O., Svátek, V.: The Ten-Year OntoFarm and its Fertilization within the Onto-Sphere. J. Web Semantics **43** (2017) 46–53
7. Jiménez-Ruiz, E., Grau, B.C., Horrocks, I.: Is my ontology matching system similar to yours? In: 8th International Workshop on Ontology Matching. (2013) 229–230
8. Jiménez-Ruiz, E., Grau, B.C., Horrocks, I., Llavori, R.B.: Logic-based assessment of the compatibility of UMLS ontology sources. J. Biomedical Semantics **2**(S-1) (2011) S2
9. Saveta, T., Daskalaki, E., Flouris, G., Fundulaki, I., Herschel, M., Ngomo, A.C.N.: Pushing the limits of instance matching systems: A semantics-aware benchmark for linked data. In: WWW, Companion Volume. (2015)
10. Saveta, T., Daskalaki, E., Flouris, G., Fundulaki, I., Herschel, M., Ngomo, A.C.N.: Lance: Piercing to the heart of instance matching tools. In: Int'l Semantic Web Conference. (2015)
11. Faria, D., Pesquita, C., Santos, E., Palmonari, M., Cruz, I.F., Couto, F.M.: The AgreementMakerLight Ontology Matching System. In: OTM Conferences - ODBASE. (2013) 527–541
12. Faria, D., Pesquita, C., Santos, E., Palmonari, M., Cruz, I.F., Couto, F.M.: Results of AML in OAEI 2017. In: ISWC International Workshop on Ontology Matching (OM). (2017)
13. Cruz, I.F., Palandri Antonelli, F., Stroe, C.: AgreementMaker: Efficient Matching for Large Real-World Schemas and Ontologies. PVLDB **2**(2) (2009) 1586–1589
14. Rybinski, M., del Mar Roldán García, M., García-Nieto, J., Montes, J.F.A.: Dismatch results for OAEI 2016. In: 11th International Workshop on Ontology Matching (OM). (2016)
15. Rybiński, M., Montes, J.F.A.: DomESA: a novel approach for extending domain-oriented lexical relatedness calculations with domain-specific semantics. Journal of Intelligent Information Systems **49**(3) (2017) 315–331
16. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
17. Kachroudi, M., Diallo, G., Ben Yahia, S.: OAEI 2017 results of KEPLER. In: 12^{th} International Workshop on Ontology Matching. Volume 2032 of CEUR-WS. (2017) 138–145
18. Chen, X., Xia, W., Jiménez-Ruiz, E., Cross, V.V.: Extending an ontology alignment system with bioportal: a preliminary analysis. In: ISWC Posters & Demonstrations Track. (2014)
19. Fridman Noy, N., Shah, N.H., Whetzel, P.L., Dai, B., Dorf, M., Griffith, N., Jonquet, C., Rubin, D.L., Storey, M.A.D., Chute, C.G., Musen, M.A.: BioPortal: ontologies and integrated data resources at the click of a mouse. Nucleic Acids Research **37**(Web-Server-Issue) (2009)
20. Khiat, A., Benaissa, M., Belfedhal, M.A.: STRIM results for OAEI 2015 instance matching evaluation. In: 10th International Workshop on Ontology Matching. (2015) 208–215
21. Khiat, A., Mackeprang, M.: I-Match and OntoIdea results for OAEI 2017. In: 12th International Workshop on Ontology Matching. (2017) 135–137
22. Sherif, M.A., Dreßler, K., Smeros, P., Ngomo, A.N.: RADON - Rapid Discovery of Topological Relations. In: AAAI Conference on Artificial Intelligence. (2017) 175–181
23. Mohammadi, M., Atashin, A.A., Hofman, W., Tan, Y.: SANOM results for OAEI 2017. In: 12th International Workshop on Ontology Matching. (2017) 185–189
24. Annane, A., Bellahsene, Z., Azouaou, F., Jonquet, C.: Building an effective and efficient background knowledge resource to enhance ontology matching. J. Web Semantics (2018)
25. Ngo, D., Bellahsene, Z.: Overview of YAM++ - (not) yet another matcher for ontology alignment task. J. Web Semantics **41** (2016) 30–49
26. Jiménez-Ruiz, E., Meilicke, C., Grau, B.C., Horrocks, I.: Evaluating mapping repair systems with large biomedical ontologies. In: 26th Workshop on Description Logics. (2013) 246–257

Semantic Similarity: A Key to Ontology Alignment

Valerie Cross

Computer Science and Software Engineering Department
Miami University, Oxford, OH 45056
crossv@muohio.edu

Abstract. Many approaches to measure the similarity between concepts that exist in two different ontologies are used in the matchers of ontology alignment systems. These matchers belong to various categories depending on the context of the similarity measurement, such as lexical, structural, or extensional matchers. This paper presents a review of various forms of semantic similarity measures. Then it examines cross-ontological semantic similarity and how various OA systems have used these along with traditional semantic similarity measure on background knowledge sources. The use of mediating ontologies in ontology alignment also may incorporate the use of semantic similarity.

Keywords: Semantic similarity, ontological similarity, cross-ontological similarity, ontology alignment, information content, mediating ontology.

1 Introduction

Similarity measurement, an important notion to compare two different objects, determines how well they agree or match each other. Ambiguity exists in the meaning of the word similarity because of its diverse use in many contexts such as biology, statistics, and psychology. The term semantic similarity has been used to refer to measures between lexical words. Natural language processing applications, such as word sense disambiguation, text summarization, annotation, and information extraction and retrieval have used numerous such measures [Budanitsky, 1999]. The growth of the Semantic Web and the explosion of ontologies, the key knowledge representation model for the Semantic Web, has renewed interest in measuring similarity. In this paper, the context is ontology alignment (OA). An ontological similarity measure is a special kind of semantic similarity measure that uses the structuring relationships between concepts in an ontology to determine a degree of similarity between those concepts. Semantic similarity is used to include similarity measures that use an ontology's structure or external knowledge sources to determine similarity between entities within one ontology or between two different ontologies. They have become a key to aligning ontologies in more sophisticated domains such as biomedical. Most of the better performing current OA systems use some background knowledge source or mediating ontology and semantic similarity measures to address where simple string matching or other OA similarity measures fail at producing mappings.

2 Brief Historical Overview of Semantic Similarity

About 30 years ago, distance in a semantic network was simply the number of edges in the path between two concepts [Rada 1989]. This distance is not sensitive to the depth of the edge in the network. It assumed a weight of 1 for all edges regardless of their hierarchical depth. This weakness has been the focus in [Wu and Palmer, 1994] and [Leacock and Chodorow, 1998]. Several pieces of information about the edge are used in determining the weight: its depth, density of edges at that depth, and strength of connection between parent and child nodes. Edge weights are reduced farther down the hierarchy and in dense parts of the graph where edges represent smaller distances. Various methods are used to normalize and convert distances into semantic similarity.

Another approach is based on the insight that conceptual similarity between two ontology concepts is related to the amount of information they share [Resnik, 1995]. The more shared information, the more similar they are. Information content (IC) is a measure of how specific a concept is in a given ontology, i.e., the more specific, the higher its IC. In [Resnik, 1995], IC is calculated relative to a selected corpus and uses a logarithmic function of the probability of the concept determined by its frequency of occurrence in the corpus. Another method [Seco et al., 2004] uses the ontology structure itself as a statistical resource; it needs no external corpus. An ontology is assumed to be organized in a meaningful, structured way; the more descendants a concept has, the less information it expresses. A concept's IC is a logarithmic function of its number of descendants and the maximum number of concepts in the ontology. In [Resnick, 1995] similarity between two concepts is a function of the IC of the most specific ancestor to both concepts in the hierarchy, i.e., their shared information. In [Jiang and Conrath, 1997] [Lin, 1998] semantic similarity is determined as a function of both the IC of each individual concept with the amount of shared IC of the two concepts.

Semantic similarity measures can also be determined from a set of features for each concept. The parameterized ratio model of similarity [Tversky 1977] uses the ratio between the cardinality of the intersection of their two sets and the sum of the cardinality of this intersection and their set symmetric difference. Parameters on the set differences depend on which concept is to be emphasized as the reference concept. The Jaccard similarity measure [Jaccard, 1901] weights both set differences by 1 to produce the ratio of the intersection cardinality over the union cardinality of the two sets. A detailed discussion of semantic similarity is presented in [Cross, 2009].

3 Measuring Concept Similarity between Different Ontologies

OA research has typically focused on finding equivalences between two concepts in different ontologies. OA techniques vary greatly depending both on what features, i.e., the schema, its instances, etc. and on what background knowledge sources such as vocabularies or other ontologies, already existing alignments, free text and search engines are used to determine the mappings [Shvaiko and Euzenat, 2013].

Early on string edit distances between the concept labels were used by OA systems. Later research developed more sophisticated similarity for use in OA matchers. The foundation for many of OA matchers can be found in [Rodriguez and Egenhofer, 2003] which relies on Tversky's parameterized ratio model of similarity. These matchers belong to various categories depending on the context of the similarity measurement, such as lexical, structural, or extensional matchers [Sabou et al., 2008]. Determining ontological similarity between entity classes a and b uses a matching process over several different sets: synonym sets (w), semantic neighborhoods (n), and distinguishing features (u). Distinguishing features are further classified into parts, functions and attributes. The similarity formula is the same for each set and given as

$$S_{\text{set-type}}(a,b)=[|A\cap B|] / [|A\cap B| + \alpha(a,b)\,|A - B| + (1 - \alpha(a,b))\,|B - A|] \text{ for } 0 \leq \alpha \leq 1$$

where A and B are description sets for entity classes a and b and are specified by the set-type = w, u, and n. The only variation to Tversky's parameterized ratio model is the setting of α which determines the importance of the non-common characteristics between a and b. The α parameter is simply determined from the depth of the entities within their respective ontologies. The parameter α is set to the ratio of the depth of a over the sum of the depths of a and b. Using α parameter gives priority to the more salient entity, i.e., the one with the greater depth. The overall similarity assessment of a and b is based on a weighted aggregation of the individual matching components S_w, S_n, and S_u. Aggregation weights depend on the assessment of the importance of each semantic component of the ontologies. Many current OA systems use this approach with their matchers for various components or features of entities and then weight the similarity of the individual matcher results either manually or through learning methods. Some OA systems employ methods to automatically weight the matchers based on an overall assessment of ontology similarity over these various kinds of sets [Pirro and Talia, 2010] [Wang et al 2010].

4 Semantic Similarity in OA Systems

OA systems have used various semantic similarity measures in a single ontology viewed as background knowledge source such as a thesaurus or mediating ontology. Concepts from the source and target ontologies are mapped into the background knowledge source. The following OA systems presented in order of their appearance in the research literature have been described in [Cross et al., 2012] in more detail. Recent OA systems use variations of semantic similarity seen in these earlier systems.

OLA [Euzenat and Valtchev, 2003] uses the lexical similarity between a pair of concept identifiers based on a set of terms for each identifier. Pairs of terms for each identifier are located in WordNet. Their term similarity is calculated using a modified Wu-Palmer measure. An aggregated similarity of proximity over all pairs of terms is calculated. **iMapper** [Su et al., 2004] increases the similarity between two concepts based on their distance in WordNet. The concepts are found in WordNet using their labels. If two terms belong to the same WordNet synset, the path distance is 1. Otherwise, the path length from each sense of one to each sense of the other is found (Rada distance). The minimum of these lengths is the semantic distance between them. If no path is found between them, they are unrelated and their similarity is not increased. **SAMBOdtf** [Lambrix et al., 2008] has the WordNet matcher that finds synonyms for concepts. If the concepts are not synonyms to the same WordNet concept, the hypernym relationships between concepts is used to determine their similarity. A domain matcher uses the UMLS. If both the source and target concepts are a synonym of the same UMLS concept, then the domain knowledge matcher sets the similarity to 0.99; otherwise the similarity is set to 0. **ASMOV** [Jean-Mary and Kabuka, 2008] checks if strings are not identical for concept labels and if available, uses WordNet or UMLS. Their lexical similarity is set to 0.99 if one label string is a synonym of the other. If one is an antonym of the other, it is set to 0. If neither and both string labels are in WordNet, it is set to Lin semantic similarity measure between the two.

CIDER [Gracia and Mena, 2008] uses a modified version of a sense semantic similarity measure to evaluate similarity between possible senses of a keyword and its synonyms to disambiguate. Semantic similarity in the filtering of mappings is adapted from the PowerMap WordNet based algorithm [Lopez et al. 2006]. The Wu-Palmer measure is used. A directional similarity is used. The validity of a mapping between concepts A and B is determined in both directions, B to A and A to B. The similarity measure is binary and is a 1 if either direction similarity is a 1 and is based on commonality between the synsets of each concept. **UFOme** [Pirro and Talia, 2010] uses a set of matchers; many have been previously developed for numerous OA systems and integrated into UFOme. The strategy predictor creates a mapping strategy by selecting and ordering the matching components. One of its matchers, the WordNet matcher, is similar to ASMOV's matcher. It uses the Lin similarity between synsets of concept terms when they do not map to the identical lexical concept in WordNet.

Using a mediating ontology is similar to using a background knowledge source. The difference if the OA system must use a simple matcher to quickly map source and target concepts to the mediating ontology, typically domain specific. Both the source and target ontologies are efficiently aligned to the mediating ontology O_M to produce a set of mappings M_{SM} and M_{TM}, respectively. In [Gross et al., 2011] [Cruz et al., 2011] a set of mediated mappings M_{ST} is created based on an exact match on the concept in O_M both the source and target concepts map to. These mediated mappings may be used when the OA process has not found direct mappings between the concepts in the two ontologies. The Uberon ontology has been used as the mediating ontology in GOMMA and AgreementMaker. An issue is both source and target concepts must map to the identical concept in the mediating ontology. The Mediating Matcher with Semantic Similarity (MMSS) was added as a new matcher to use semantic similarity measures between the mapped concepts in O_M even if no exact match exists [Cross et al., 2012].

5 Conclusions and Possible Future Directions

Semantic similarity has been reviewed and its important role in the ontology alignment task has been emphasized. Tversky's parameterized ratio model of similarity has been discussed as fundamental to developing similarity measures between concepts in different ontologies [Rodriguez and Egenhofer, 2003]. Although semantic similarity measures are essential to the OA task, more research needs to be done to determine if specific ones have better performance. The OA task should be used as a benchmark for performance evaluations on existing and new measures.

References

1. Budanitsky, A.: Lexical Semantic Relatedness and Its Application in Natural Language Processing, Computer Systems Research Group, Tech Report, University of Toronto (1999)
2. Cross, V., Silwal, P., Morell, D.: Using a Reference Ontology with Semantic Similarity in Ontology Alignment. Proc. of Int. Conf.on Biomedical Ontologies (ICBO), Graz (2012)
3. Cross, V.: Ontological Similarity. In Data Mining in Biomedicine Using Ontologies, Norwood. MA:Artech House, ISBN-13:978-1-59693-370-5, pp. 23-43 (2009)
4. Cruz, I. F., Stroe, C., Caimi, F., Fabiani, A., Pesquita, C., Couto, F. M., Palmonari, M.: Using AgreementMaker to Align Ontologies for OAEI 2011. OM Workshop, ISWC (2011)
7. Euzenat J. and Valtchev, P.: An integrative proximity measure for ontology alignment. Proc.ISWC-2003 Workshop Semantic Information integration, (FL US), pp. 33–38, (2003)
8. Gracia, J., Mena, E.: Ontology Matching with CIDER: Evaluation Report for the OAEI 2008, Proc. 3rd Ontology Matching worshop, Karlruhe (DE), pp140-146 (2008)
9. Gross, A., Hartung, M., Kirsten, T., and Rahm, E.: Mapping Composition for Matching Large Life Science Ontologies. In Proc. of Int. Conf. on Biomedical Ontologies, pp 109–116 (2011)
10. Jaccard, P., Étude comparative de la distribution florale dans une portion des Alpes et des Jura. Bulletin del la Société Vaudoise des Sciences Naturelles, Vol. 37, pp. 547-579 (1901)
11. Jean-Mary, Y.R., Kabuka, M.R.: ASMOV: Results for OAEI 2008, OM Workshop (2008)
12. Jiang J. and Conrath D.: Semantic similarity based on corpus statistics and lexical taxonomy. In: Proc. of the 10th Int. Conf. on Research on Computational Linguistics, Taiwan (1997)
13. Lambrix, P., Tan, He, Liang, Qiang: SAMBO and SAMBOdtf Results for the Ontology Alignment Evaluation Initiative, 3rd International Workshop on Ontology Matching (2008)
14. Leacock, C. and Chodorow M.: Combining local context and WordNet similarity for word sense identification. In WordNet: An Electronic Lexical Database, pp. 265-283, Cambridge, MA: The MIT Press (1998)
15. Lin. D.: An Information-theoretic Definition of Similarity. Proc. 15th International Conference on Machine Learning, Madison, Wisconsin, July 1998, pp. 296-304 (1998)
16. Lopez, V., Sabou, M., and Motta, E.: Powermap: Mapping the real semantic web on the Fly. In Proc. of 5th International Semantic Web Conference, Athens, GA (2006)
17. Pirro, G., Talia D.: UFOme: an ontology mapping system with strategy prediction capabilities. Data Knowl.Eng. 69.5, 444-471 (2010)
18. Rada R, Mili H, Bicknell E, Blettner M: Development and application of a metric on semantic nets. In: IEEE Transaction on Systems, Man, and Cybernetics. 19. pp 17–3 (1989)
19. Resnik, P.: Semantic Similarity in a Taxonomy: An Information-Based Measure and its Application to Problems of Ambiguity in Natural Language. Journal of Artificial Intelligence Research Vol. 11, pp. 95-130 (1999)
20. Rodriguez, M. A., Egenhofer, M. J.: Determining Semantic Similarity among Entity Classes from Different Ontologies. IEEE Transactions on Knowledge and Data Engineering Vol 15, Issue 2(2003): p 442-456 (2003).
21. Sabou, M., d'Aquin, M., Motta, E.: Exploring the Semantic Web as Background Knowledge for Ontology Matching. J. Data Semantics 11: pp. 156-190 (2008)
22. Seco N, Veale T, Hayes J.: An intrinsic information content metric for semantic similarity in WordNet. In: ECAI. pp 1089–1090 (2004)
23. Shvaiko, P. and Euzenat, J.: Ontology Matching: State of the Art and Future Challenges. IEEE Trans. Knowl. Data Eng. 25(1): pp. 158-176 (2013)
24. Su, Xiaomeng: Semantic Enrichment for Ontology Mapping. Ph.D. Thesis, Dept. of Computer and Information Science, Norwegian University of Science and Technology (2004)
25. Tversky, A.: Features of Similarity. Psychological Rev., 84, pp. 327--352 (1977)
26. Wu, Z. and Palmer, M: Verb semantics and lexical selection. Proc. 32nd Annual Meeting of the Association for Computational Linguistics, Las Cruces, New Mexico, pp 133-138 (1994)

Complex matching based on competency questions for alignment: a first sketch

Elodie Thiéblin, Ollivier Haemmerlé, Cassia Trojahn

IRIT & Université de Toulouse 2 Jean Jaurès, Toulouse, France
{`firstname.lastname`}`@irit.fr`

1 Introduction

A complex alignment between a source ontology o_1 and a target ontology o_2 is a set of correspondences with at least a complex correspondence. Complex correspondences (e.g., o_1*:GenusRank* $\equiv \exists$ o_2*:hasRank.*{o_2*:genus*}) involve logical constructors (e.g., property restriction) or transformation functions of literal values (e.g., string concatenation). Complex matching approaches have emerged in the literature in the last years [10, 8, 13, 6]. While some rely on statistical methods [8, 13], others rely on linguistic matching conditions [10] or knowledge rules [6]. Many of them are based on correspondence patterns [10, 8, 13]. Following a different approach, this paper proposes a complex matching approach which relies on the notion of *Competency Question for Alignment* (CQA). CQAs express the knowledge that an alignment should cover. As for ontology authoring, they take the form of NLP questions or SPARQL queries. Our approach takes as input a set of CQAs translated into SPARQL queries over the source ontology. The answer to each query is a set of instances retrieved from a knowledge base described by the source ontology. These instances are matched with those of a knowledge base described by the target ontology. The generation of the correspondence is performed by matching the graph-pattern from the source query to the lexically similar surroundings of the target instances. For example, given the source query `SELECT ?x WHERE` {`?x a` o_1`:GenusRank.`}, and an output correspondence o_1*:GenusRank* $\equiv \exists$ o_2*:hasRank.*{o_2*:genus*}, one could translate the source query into `SELECT ?x WHERE` {`?x` o_2`:hasRank` o_2`:genus.`}. Our approach was evaluated on a set of four knowledge bases about plant taxonomy.

2 Competency questions for alignment

In ontology matching system design, a question that rises is "Are there any specifications to the matching process ? If so, what are the needs/requirements that an alignment should meet ?". Few guidelines in the literature are given to characterise an alignment and/or the matching process. One of the few examples is the NeOn methodology [4], which characterises both alignment and matching process through a set of questions: i) is matching performed under time constraints ? ii) has matching to be performed automatically ? iii) must the alignment be correct ? complete ? and iv) what type of operation (merging, query, etc.) is to be performed ? Through these questions, qualitative and applicative characteristics of an alignment and the matching process are defined. However, they do not help specifying the knowledge the alignment should cover, i.e. its scope. Here, we extend the notion of "needs" for the alignment as defined in [4] by proposing the notion of *Competency Question for Alignment* (CQA).

In order to formalise the knowledge needs of an ontology, *competency questions* (CQ) have been introduced as *ontology's requirements in the form of questions the ontology must be able to answer* [5]. Here, a CQA expresses the knowledge that an alignment should cover in the best case (if both ontologies' scope can answer the CQA). The first difference between CQA and CQ in ontology authoring is that the scope of the CQA is limited by the intersection of its source and target ontologies' scopes. The second difference is that this maximal and ideal alignment's scope is not known *a priori* (as it is the purpose of the alignment). Measuring the completeness or the competency of an alignment is, however, out of the scope of this work.

Taking into account the characteristics of CQs in the literature, we adapt them for CQAs. In [9], the authors define a set of CQ characteristics (question type, element visibility, question polarity, predicate arity, modifier, domain independent element), as well as a set of competency question patterns. Inspired from the predicate arity in [9], we introduce the notion of **question arity**, which represents the arity of the expected answers to a CQA:

- A *unary* question expects a set of instances or values, e.g., "What are the genus taxa?" *(Triticum), (Anas).*
- A *binary* question expects a set of instances or value pairs, e.g., "What is the rank of a taxon?" *(Plantae, Kingdom), (Triticum, Genus).*
- A *n-ary* question expects a tuple of size 3 or more, e.g., "In which classification is the rank of a taxon defined?" *(Triticum, Genus, Linnaeus 1753), (Plantae, Kingdom, Haeckel 1866).*

Concerning the use of CQAs, they can be used for both alignment evaluation by verifying that an alignment covers a user-defined scope, as in the OA4QA task [12], and for guiding alignment creation. Our approach falls in the latter case.

3 Proposed approach

The approach takes as input a set of CQAs translated into SPARQL queries over the source ontology. The answer to each input query is a set of instances, which are matched with those of a knowledge base described by the target ontology. The matching is performed by finding the lexically similar surroundings of the target instances. Here, CQAs are limited to unary questions, (class expressions, set of instances expected), of selection type, polarity positive and no modifier. The approach is developed in 11 steps, as depicted in Figure 1:

① Extract source DL formula e_s from SPARQL CQA (e.g., o_1*:Genus*)

② Extract lexical information from the CQA, L_s set labels of atoms from the DL formula (e.g., "Genus", "genre")

③ Extract source instances $inst_s$ (e.g., o_1*:triticum*)

④ Find equivalent or similar (same label) target instances $inst_t$ to the source instances $inst_s$ (e.g. o_1*:triticum* $\sim$ o_2*:wheat*)

⑤ Retrieve description of target instances: set of triples and object/subject type (e.g. ⟨*(o_2:wheat, o_2:genus) : o_2:hasRank, o_2:genus: o_2:Rank*⟩, ⟨*(o_2:emmer_wheat, o_2:wheat) : o_2:hasHigherTaxon, o_2:emmer_wheat: o_2:Taxon*⟩)

⑥ For each triple, retrieve L_t labels of entities (e.g., o_2*:hasRank* → "taxonomic rank", o_2*:genus* → "genus", o_2*:Rank* → "rank")

⑦ Compare L_s and L_t using a string comparison metric (e.g., Levenshtein distance with a threshold)

⑧ Keep the triples with the summed similarity of their labels above a threshold τ. Keep the object(/subject) type if its similarity is better than the one of the object(/subject). (e.g. sim(o_2*:genus*, L_s) > sim(o_2*:Rank*,L_s) so we only keep o_2*:genus* in the triple)

⑨ Express the triple into a DL formula (e.g., ∃ o_2*:hasRank*.{o_2*:genus*})

⑩ Aggregate the formulas into an explicit or implicit form: if two DL formulas have a common atom in their right member (target member): the atoms which differed are put together (e.g., ∃ o_2*:hasRank*.{o_2*:genus*} and ∃ o_2*:hasRank*.{o_2*:kingdom*} would give 2 formulae: ∃ o_2*:hasRank*.{o_2*:genus*, o_2*:kingdom*} and ∃ o_2*:hasRank*.⊤)

⑪ Put e_s and e_t together in a correspondence (e.g., o_1*:GenusRank* ≡ ∃ o_2*:hasRank*.{o_2*:genus*}) and express this correspondence in EDOAL

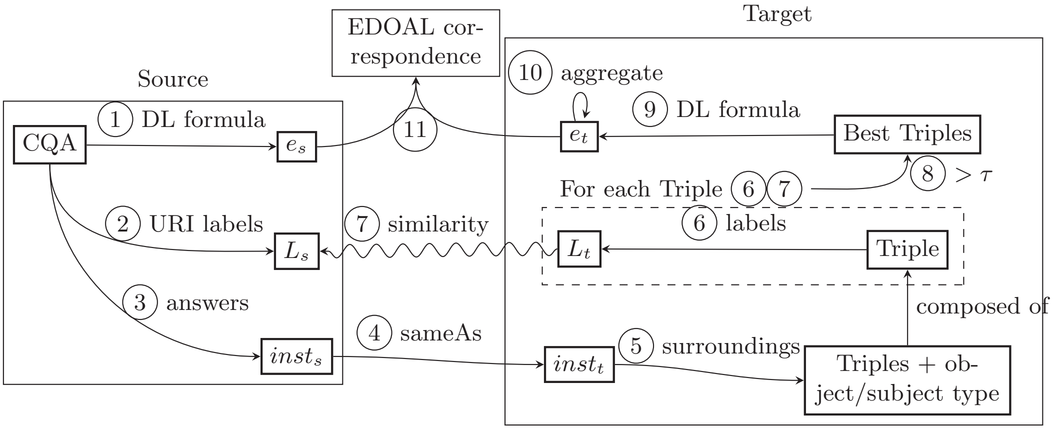


Fig. 1: Schema of the general approach.

4 Evaluation

We evaluated our approach on a set of four knowledge bases about plant taxonomy: AgronomicTaxon [11], Agrovoc [3], TaxRef-LD [7], and DBpedia [2]. All except AgronomicTaxon contain thousands of taxa ($\sim 32,000$ for Agrovoc, $\sim$

500, 000 for TaxRef-LD, $\sim 307,000$ for DBpedia). Their instances are linked with *skos:exactMatch*, *skos:closeMatch*, *owl:sameAs* and *rdfs:seeAlso*. Two CQAs were used in the evaluation i) What are the genus taxa ? ii) What are the taxa ? Each CQA was manually translated into a SPARQL query for each ontology. All the source-target combinations of ontologies were tested, resulting in 12 alignment pairs for each CQA. For each pair, the output correspondences were manually evaluated. A correspondence was considered correct if their members are semantically equivalent. The evaluation metrics are i) precision: number of correct output correspondences / number of output correspondences and ii) top-k accuracy, as used in the evaluation of [1]: number of CQAs per pair for which at least a correct correspondence was output. As we do not compare our alignments to a reference alignment (because one would not cover all possible complex correspondences), we cannot compute recall. Table 1 presents, for each pair of ontologies and for each CQA, the number of correct correspondences out of the total number of correspondences generated by the approach. The overall precision is 32.8% (44/134) and the top-k accuracy is 83.4% (20/24). When the ontologies have a similar structure, we obtain a better precision (Agrovoc – TaxRef-LD).

	Source/Target	AgronomicTaxon	Agrovoc	TaxRef-LD	DBpedia
Genus	AgronomicTaxon		1 / 1	3 / 3	2 / 15
	Agrovoc	1 / 3		3 / 5	2 / 8
	TaxRef-LD	1 / 6	1 / 2		3 / 10
	Dbpedia	1 / 1	1 / 2	4 / 6	
Taxa	AgronomicTaxon		**0 / 4**	4 / 4	4 / 21
	Agrovoc	2 / 4		4 / 12	3 / 18
	TaxRef-LD	1 / 6	1 / 2		2 / 8
	DBpedia	**0 / 4**	**0 / 1**	**0 / 4**	

Table 1: Number of correct / number of output correspondences per CQA.

Some found correspondences were totally wrong, such as “a taxon in Agrovoc (a concept having a taxonomic rank) is something which has been represented by a statue in Wikidata” (for sake of comprehension, we express the correspondences in natural language). Other found correspondences were not precise enough such as “a taxon in Agrovoc is something having a taxon below it in a taxonomy in AgronomicTaxon”, which would be correct with a subsumption relation. For some CQAs, more than one correspondence were evaluated as correct. The first reason is that some axioms of the ontology are equivalent (inverse properties, etc.). The second one is that the knowledge bases sometimes import other ontologies and instances. For example, TaxRef-LD imports data from Agrovoc, VTO and NCBI. Hence, they share common elements. Finally, as Table 1 shows, the Taxa CQA with DBpedia as source ontology does not output any correct correspondence because a taxon in DBpedia is an instance of the *dbo:Species* class. The source SPARQL query only contains this URI. Therefore, the query labels on which the lexical similarity is based are those of *dbo:Species* which do not contain anything related to *Taxon*. Most the correspondences found for this query represent the taxa having specy as taxonomic rank.

5 Conclusion and perspectives

This paper introduced the notion of competency questions for alignment (CQAs) and proposed a complex matching approach guided by CQAs. As the approach relies on the labels from the SPARQL query, the similarity of the ontologies' lexical layers impacts the output correspondences. As perspectives, we plan to perform the instance matching phase using key detection techniques, to use more linguistic evidence in the matching process, to consider binary CQAs, and work on the semantics of the confidence of complex correspondences.

Acknowledgements We would like to thank Catherine Roussey and Nathalie Hernandez for their contribution on the Agronomic dataset.

References

1. An, Y., Hu, X., Song, I.Y.: Learning to discover complex mappings from web forms to ontologies. In: ACM Conference on Information and knowledge management. pp. 1253–1262 (2012)
2. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.: Dbpedia: A nucleus for a web of open data. The semantic web (2007)
3. Caracciolo, C., Stellato, A., Morshed, A., Johannsen, G., Rajbhandari, S., Jaques, Y., Keizer, J.: The agrovoc linked dataset. Semantic Web 4(3), 341–348 (2013)
4. Euzenat, J., Le Duc, C.: Methodological guidelines for matching ontologies. In: Ontology engineering in a networked world, pp. 257–278. Springer (2012)
5. Grüninger, M., Fox, M.S.: Methodology for the design and evaluation of ontologies. international joint conference on artificial inteligence. In: Workshop on Basic Ontological Issues in Knowledge Sharing. vol. 15, p. 34 (1995)
6. Jiang, S., Lowd, D., Kafle, S., Dou, D.: Ontology matching with knowledge rules. In: Transactions on Large-Scale Data-and Knowledge-Centered Systems XXVIII, pp. 75–95. Springer (2016)
7. Michel, F., Gargominy, O., Tercerie, S., Faron-Zucker, C.: A Model to Represent Nomenclatural and Taxonomic Information as Linked Data.Application to the French Taxonomic Register, TAXREF. In: S4BioDiv (2017)
8. Parundekar, R., Knoblock, C.A., Ambite, J.L.: Discovering concept coverings in ontologies of linked data sources. In: ISWC. pp. 427–443. Springer (2012)
9. Ren, Y., Parvizi, A., Mellish, C., Pan, J.Z., van Deemter, K., Stevens, R.: Towards Competency Question-Driven Ontology Authoring. In: The Semantic Web: Trends and Challenges, vol. 8465, pp. 752–767 (2014)
10. Ritze, D., Völker, J., Meilicke, C., Šváb Zamazal, O.: Linguistic analysis for complex ontology matching. In: 5th workshop on ontology matching. pp. 1–12 (2010)
11. Roussey, C., Chanet, J.P., Cellier, V., Amarger, F.: Agronomic taxon. In: Proceedings of the 2nd International Workshop on Open Data. p. 5. ACM (2013)
12. Solimando, A., Jiménez-Ruiz, E., Pinkel, C.: Evaluating ontology alignment systems in query answering tasks. In: ISWC Posters & Demos. pp. 301–304 (2014)
13. Walshe, B., Brennan, R., O'Sullivan, D.: Bayes-recce: A bayesian model for detecting restriction class correspondences in linked open data knowledge bases. International Journal on Semantic Web and Information Systems 12(2), 25–52 (2016)

A Proposal for Optimizing Internetwork Matching of Ontologies

Fabio Santos, Kate Revoredo* and Fernanda Baião**

Department of Applied Informatics
Federal University of the State of Rio de Janeiro (Unirio), Rio de Janeiro, Brazil
{fabio.santos, katerevoredo, fernanda.baiao}@uniriotec.br

Abstract. A System-of-Systems (SoS) is a set of independent information systems that must communicate with each other towards providing a specific service. Therefore, effectively integrating these systems is demanding. Considering that each system is conceptually described by a unique ontology, the conceptual support for the whole SoS demands the alignment of all ontologies, deriving a network of ontologies. Existing ontology matching techniques may be used for the task; however, due to the recently increasing size of the ontologies and the potential number of ontologies being aligned, current approaches may suffer from scalability and performance issues. In this paper, we introduce an approach to reduce the number of potential correspondences, therefore optimizing the process of creating a network of ontologies. A preliminary experiment was conducted, showing the potential of the proposed approach.

Keywords: network of ontologies · network matching · data integration

1 Introduction

A System-of-Systems (SoS) is defined as a set of independent information systems (IS), providing functionalities derived from the interoperability among them [2]. The development and research on SoS have been gaining increasing attention due to the relevance of several domains such as smart cities, health, emergency response systems, and crisis management systems [6]. Considering that each IS within a SoS is conceptually described by a unique ontology describing its domain, the conceptual support for the whole SoS demands the interoperation of its composing IS, thus requiring the alignment of all the corresponding ontologies. Moreover, a single SoS may embrace several domains, thus requiring by itself a *network of ontologies* as its conceptual support. Therefore, there is an increasing need for aligning networks of ontologies, a problem called *internetwork matching.*

Traditional solutions for ontology matching may be applied for solving the internetwork matching problem, either using a pairwise or a holistic strategy [8]. While the former interactively matches one pair of ontologies from different

---

* Partially funded by Unirio (PQ-UNIRIO N01/2018)

** Partially funded by CNPq (401505/2014-6)

networks at a time, the latter considers all the network at once. In both cases, all pairs of entities from each ontology that composes the networks are analyzed, which poses a severe restriction in terms of scalability since the required number of comparisons for computing the alignments grows exponentially to the number and size of each ontology. Therefore, there is a need for optimized solutions to this problem [9].

This work proposes an optimized approach for the internetwork matching challenge [1] that tries to reduce the number of pairs to be evaluated during the matching process, thus avoiding unneeded computation while preserving the alignment quality. We evaluated the proposed approach in a preliminary experiment using an OAEI dataset.

This work is organized as follows: Section 2 defines the internetwork matching problem. Our proposed approach is detailed in Section 3. Preliminary evaluation results are in Section 4. Finally, section 5 concludes and points to future work.

2 Problem Definition

A network of ontologies is formally defined as $\Gamma =< \Omega, \Lambda >$, where Ω is a finite set of ontologies and $\Lambda(O, O')$ is a set of alignments between pairs of ontologies belonging to Ω [5]. Given a set of two or more networks of ontologies $\Psi = \{\Gamma_1, \Gamma_2, ..., \Gamma_n\}$, the internetwork matching problem searches for a final network of ontologies Γ_f resulting from the alignments of the networks in Ψ. For instance, Figure 1 depicts two networks of ontologies, each one with 3 ontologies, describing two Systems-of-Systems. The goal is to match these two networks, finding a unique network of ontologies.

One of the approaches for matching networks of ontologies is pairwise. Given a set of networks of ontologies, the pairwise internetwork matching sequentially computes the alignment of each pair of ontologies from each pair of networks from this set. For example, given two networks of ontologies $\Gamma =< \Omega, \Lambda >$ and $\Gamma' =< \Omega', \Lambda' >$, in which $\Omega = \{O_1, O_2\}$ and $\Omega' = \{O_3\}$, the pairwise internetwork matching is obtained by computing $(((O_1 \times O_2) \cup (O_1 \times O_3)) \cup (O_2 \times O_3))$. That is, the pairwise internetwork matching approach computes all matchings between all pairs of ontologies inside each network that is being aligned.

Networks frequently have isomorphisms and trivial alignments that may cause the pairwise approach to find the same alignments more than once, thus requiring an additional step to merge the resulting matches at the end. In the case of isomorphisms, identical correspondences between same entities may be generated (for instance, in Figure 1 a matcher tool may find $A_{1,1'} = \{< O_1.a_1, O'_1.a^{'}_1, => , < O_1.b_1, O'_1.b^{'}_1, =>\}$). The case of a trivial alignment occurs when a group of entities, that was previously aligned in a network, appears in another network. For instance, a pairwise matcher that receives the networks Γ and Γ' and has previously computed the intra-network alignments $A_{1,2} =< O_1.b_1, O_2.d_2, \sqsubseteq>$ and $A_{1',2'} =< O'_1.b^{'}_1, O'_2.d^{'}_2, \sqsubseteq>$ will work unnecessarily to produce $A_{1,2'} =< O_1.b_1, O'_2.d^{'}_2, \sqsubseteq>$ and $A_{1',2} =< O'_1.b^{'}_1, O_2.d_2, \sqsubseteq>$, which are trivial alignments.

The main weakness of pairwise approach is the number of comparisons needed to compute all alignments and the lack of ability to handle isomorphisms and intra-network alignments.

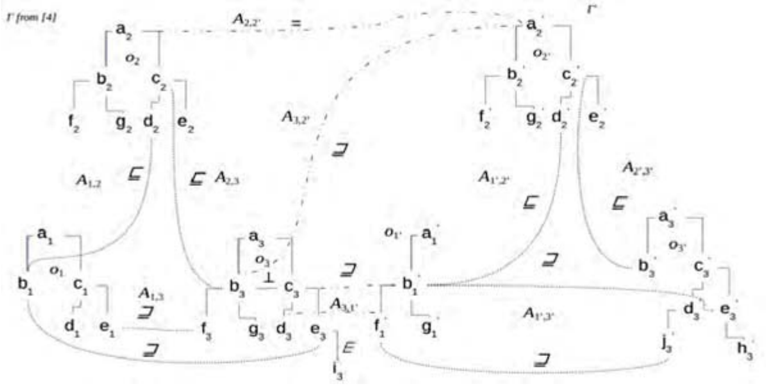


Fig. 1: *Internetwork matching of networks Γ and Γ'.*

3 Proposed Approach

Given the limitations discussed in Section 2, we propose SubInterNM, a new subsumed approach for the Internetwork matching problem. SubInterNM avoids unnecessary computation by identifying and reusing trivial and subsumed alignments already computed in the networks of ontologies that are being aligned. In such cases, as the intersection among the networks becomes larger, the set of evaluated correspondences during the alignment process tends to get smaller.

For instance, consider the example depicted in Figure 1. The networks Γ and Γ' were aligned using an internetwork matcher. Since O_2 and O'_2 are identical, they do not need to be exhaustively compared. Also, both pairs of ontologies O_1 and O'_1 and O_3 and O'_3 share some subset of entities, thus common parts could be eliminated to compute the final network of ontologies resulted from the internetwork matching problem scenario.

Casanova et al. [3] proposed operations over lightweight ontologies. They define lightweight ontologies as ontologies restricted to DL-Lite core with arbitrary number restrictions. They use a MEG (minimal equivalent graph) approach to create a constraint graph in polynomial time if the graph is acyclic. If the graph is complete, then the problem is NP-Hard. To avoid that, a normalization step is conducted to simplify the graph structure and keep them lightweight.

Our proposed approach SubInterNM uses a combined set of lightweight operations from [3] to verify the existence of isomorphisms. We extrapolate the original idea to use in networks environments, instead of just single ontologies.

4 Preliminary Results

Our proposed approach was evaluated in a preliminary experiment using ontologies from the OAEI conference dataset. We experimented 5 distinct scenarios, and in each of them we specified two networks of ontologies to be aligned. The experiments were defined by increasing the number of ontologies in the network, as well as varying the ontologies and the number of common ontologies between the networks, in order to assess how well our approach would handle the existence of trivial and subsumed alignments:

- 2x2: $\Omega = \{\text{conference, cmt}\}$ and $\Omega' = \{\text{cmt, sigkdd}\}$;
- 3x2: $\Omega = \{\text{conference, cmt, ekaw}\}$ and $\Omega' = \{\text{cmt, sigkdd}\}$;
- 3x3: $\Omega = \{\text{conference, cmt, ekaw}\}$ and $\Omega' = \{\text{cmt, sigkdd, conference}\}$;
- 4x3: $\Omega = \{\text{conference, cmt, dblp, ekaw}\}$ and $\Omega' = \{\text{cmt, sigkdd, conference}\}$;
- 4'x3: $\Omega = \{\text{conference, cmt, edas, ekaw}\}$ and $\Omega' = \{\text{cmt, sigkdd, conference}\}$.

In order to compare our proposed SubInterNM approach against the pairwise internetwork matching approach, we implemented the pairwise approach using the existing matching system ALIN [4] in all experiments. ALIN was selected due to the good results achieved on OEAI 2017, and due to our access to the code [10]. To use ALIN as a *blackbox* (i.e., without having to change its code), for each internetwork matching experiment, we built all the pairs of ontologies to be aligned and interactively invoked ALIN. SubInterNM was implemented using operations defined by Casanova et al. [3], initially considering only the isomorphisms. After, the ALIN matching system[4] was also invoked to compute the alignments between the results from Ω and $\Omega^{'}$. So we computed:

Ω as $O_1 \cup O_2 ... \cup O_n - O_1 \cap O_1^{'} - O_1 \cap O_2^{'} - ... - O_n \cap O_n^{'}$

$\Omega^{'}$ as $O_1^{'} \cup O_2^{'} ... \cup O_n^{'} - O_1^{'} \cap O_1 - O_1^{'} \cap O_2 - ... - O_n^{'} \cap O_n$

Table 1: Total number of comparisons computed by each approach

Experiment	**Pairwise**	**SubInterNM**	**% of reduction**
2x2	14,138	5,608	60.3
3x2	22,236	10,027	54.9
3x3	38,893	27,039	30.4
4x3	42,319	27,039	36.1
4'x3	57,497	43,420	24.4

Table 1 shows the number of pairs analyzed (i.e., comparisons) by pairwise and subsumed approach to compute alignments in each experiment. It is possible to verify that SubInterNM reduces the number of comparisons needed to the matching process by at least 24%, therefore representing a successful way to deal with large networks in the internetwork matching problem.

Future work will try to reduce even more the number of comparisons with the detection of intra-network alignments. All the data gathered in the experiment is available at (https://bit.ly/2M3jIYS).

5 Conclusion

This work addressed the problem of internetwork ontology matching, a natural evolution of the classical ontology matching problem for highly interconnected scenarios of Systems of Systems, which are of increasing popularity and relevance. We proposed SubInterNM, an approach for internetwork ontology matching that optimizes the required computation of correspondences by identifying and reusing trivial and subsumed alignments. Preliminary evaluation results showed the potential of the approach and opportunities for improvement, in scenarios using lightweight ontologies. The computation of subsumed networks posed an overhead computation since it is a well-known NP-Hard problem [7]. Therefore, further implementations may compute trivial alignments, using it as background knowledge for the matching process [9].

In future work, we expect to improve our implementation, including parallel programming and infrastructure. We also plan to move forward exploring trivial alignments in newer versions of the tool.

References

1. de Abreu Santos, F.M., Revoredo, K., Baião, F.A.: Paving a research roadmap on network of ontologies. In: Proceedings of the 12th International Workshop on Ontology Matching co-located with the 16th International Semantic Web Conference (ISWC 2017), Vienna, Austria, October 21, 2017. pp. 221–222 (2017), `http://ceur-ws.org/Vol-2032/om2017_poster8.pdf`
2. Boehm, B.: A view of 20th and 21st century software engineering. In: Proceedings of the 28th international conference on Software engineering. pp. 12–29. ACM (2006)
3. Casanova, M.A., de Macêdo, J.A., Sacramento, E.R., Pinheiro, Â.M., Vidal, V.M., Breitman, K.K., Furtado, A.L.: Operations over lightweight ontologies. In: OTM Confederated International Conferences" On the Move to Meaningful Internet Systems". pp. 646–663. Springer (2012)
4. Da Silva, J., Baiao, F.A., Revoredo, K., Euzenat, J.: Semantic interactive ontology matching: synergistic combination of techniques to improve the set of candidate correspondences. In: OM 2017-12th ISWC workshop on ontology matching. pp. 13–24. No commercial editor. (2017)
5. Euzenat, J.: Revision in networks of ontologies. Artificial intelligence **228**, 195–216 (2015), `ftp://ftp.inrialpes.fr/pub/exmo/publications/euzenat2015a.pdf`
6. Fitzgerald, J., Foster, S., Ingram, C., Larsen, P.G., Woodcock, J.: Model-based engineering for systems of systems: the compass manifesto. COMPASS Interest Group, Tech. Rep. Manifesto Version **1** (2013)
7. Hsu, H.T.: An algorithm for finding a minimal equivalent graph of a digraph. Journal of the ACM (JACM) **22**(1), 11–16 (1975)
8. Megdiche, I., Teste, O., Trojahn, C.: An extensible linear approach for holistic ontology matching. In: International Semantic Web Conference. pp. 393–410. Springer (2016)
9. Shvaiko, P., Euzenat, J.: Ontology matching: state of the art and future challenges. IEEE Transactions on knowledge and data engineering **25**(1), 158–176 (2013)
10. da Silva, J., Baiao, F.A., Revoredo, K.: Alin results for oaei 2017. In: OM-2017: Proceedings of the Twelfth International Workshop on Ontology Matching. p. 114 (2017)

Results of the
Ontology Alignment Evaluation Initiative 2018*

Alsayed Algergawy[1], Michelle Cheatham[2], Daniel Faria[3], Alfio Ferrara[4],
Irini Fundulaki[5], Ian Harrow[6], Sven Hertling[7], Ernesto Jiménez-Ruiz[8,9],
Naouel Karam[10], Abderrahmane Khiat[11], Patrick Lambrix[12], Huanyu Li[12],
Stefano Montanelli[4], Heiko Paulheim[7], Catia Pesquita[13], Tzanina Saveta[5],
Daniela Schmidt[14], Pavel Shvaiko[15], Andrea Splendiani[6], Elodie Thiéblin[16],
Cássia Trojahn[16], Jana Vataščinová[17], Ondřej Zamazal[17], and Lu Zhou[2]

[1] Friedrich Schiller University Jena, Germany
`alsayed.algergawy@uni-jena.de`
[2] Data Semantics (DaSe) Laboratory, Wright State University, USA
`{michelle.cheatham, zhou.34}@wright.edu`
[3] Instituto Gulbenkian de Ciência, Lisbon, Portugal
`dfaria@igc.gulbenkian.pt`
[4] Università degli studi di Milano, Italy
`{alfio.ferrara,stefano.montanelli}@unimi.it`
[5] Institute of Computer Science-FORTH, Heraklion, Greece
`{jsaveta,fundul}@ics.forth.gr`
[6] Pistoia Alliance Inc., USA
`{ian.harrow,andrea.splendiani}@pistoiaalliance.org`
[7] University of Mannheim, Germany
`{sven,heiko}@informatik.uni-mannheim.de`
[8] Department of Informatics, University of Oslo, Norway
`ernestoj@ifi.uio.no`
[9] The Alan Turing Institute, London, UK
`ejimenez-ruiz@turing.ac.uk`
[10] Fraunhofer FOKUS, Berlin, Germany
`naouel.karam@fokus.fraunhofer.de`
[11] Freie Universität Berlin, Germany
`abderrahmane.khiat@fu-berlin.de`
[12] Linköping University & Swedish e-Science Research Center, Linköping, Sweden
`{patrick.lambrix,huanyu.li}@liu.se`
[13] LASIGE, Faculdade de Ciências, Universidade de Lisboa, Portugal
`cpesquita@di.fc.ul.pt`
[14] Pontifical Catholic University of Rio Grande do Sul, Brazil
`daniela.schmidt@acad.pucrs.br`
[15] TasLab, Trentino Digitale SpA, Trento, Italy
`pavel.shvaiko@tndigit.it`
[16] IRIT & Université Toulouse II, Toulouse, France
`{cassia.trojahn,elodie.thieblin}@irit.fr`
[17] University of Economics, Prague, Czech Republic
`ondrej.zamazal@vse.cz`

---

* Note that the only official results of the campaign are on the OAEI web site.

Abstract. The Ontology Alignment Evaluation Initiative (OAEI) aims at comparing ontology matching systems on precisely defined test cases. These test cases can be based on ontologies of different levels of complexity (from simple thesauri to expressive OWL ontologies) and use different evaluation modalities (e.g., blind evaluation, open evaluation, or consensus). The OAEI 2018 campaign offered 12 tracks with 23 test cases, and was attended by 19 participants. This paper is an overall presentation of that campaign.

1 Introduction

The Ontology Alignment Evaluation Initiative[1] (OAEI) is a coordinated international initiative, which organizes the evaluation of an increasing number of ontology matching systems [18, 20]. The main goal of the OAEI is to compare systems and algorithms openly and on the same basis, in order to allow anyone to draw conclusions about the best matching strategies. Furthermore, our ambition is that, from such evaluations, 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) [45]. 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) [4]. From 2006 until the present, the OAEI campaigns were held at the Ontology Matching workshop, collocated with ISWC [1–3, 6–8, 11, 14–17, 19], which this year took place in Monterey, CA, USA[2].

Since 2011, we have been using an environment for automatically processing evaluations (§2.1) which was 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. Since OAEI 2017, a novel evaluation environment called HOBBIT (§2.1) was adopted for the HOBBIT Link Discovery track, and later extended to enable the evaluation of other tracks. Some tracks are run exclusively through SEALS and others through HOBBIT, but several allow participants to choose the platform they prefer.

This paper synthesizes the 2018 evaluation campaign and introduces the results provided in the papers of the participants. The remainder of the paper is organized as follows: in §2, we present the overall evaluation methodology; in §3 we present the tracks and datasets; in §4 we present and discuss the results; and finally, §5 concludes the paper.

---

[1] `http://oaei.ontologymatching.org`

[2] `http://om2018.ontologymatching.org`

[3] `http://www.seals-project.eu`

2 Methodology

2.1 Evaluation platforms

The OAEI evaluation was carried out in one of two alternative platforms: the SEALS client or the HOBBIT platform. Both have the goal of ensuring reproducibility and comparability of the results across matching systems.

The **SEALS client** was developed in 2011. It is a Java-based command line interface for ontology matching evaluation, which requires system developers 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 is provided to the participants, describing how to wrap a tool and how to run a full evaluation locally.

The **HOBBIT platform**[4] was introduced in 2017. It is a web interface for linked data and ontology matching evaluation, which requires systems to be wrapped inside docker containers and include a SystemAdapter class, then being uploaded into the HOBBIT platform [31].

Both platforms compute the standard evaluation metrics against the reference alignments: precision, recall and F-measure. In test cases where different evaluation modalities are required, evaluation was carried out *a posteriori*, using the alignments produced by the matching systems.

2.2 OAEI campaign phases

As in previous years, the OAEI 2018 campaign was divided into three phases: preparatory, execution, and evaluation.

In the **preparatory phase**, the test cases were provided to participants in an initial assessment period between June 15^{th} and July 15^{th}, 2018. The goal of this phase is to ensure that the test cases make sense to participants, and give them the opportunity to provide feedback to organizers on the test case as well as potentially report errors. At the end of this phase, the final test base was frozen and released.

During the ensuing **execution phase**, participants test and potentially develop their matching systems to automatically match the test cases. Participants can self-evaluate their results either by comparing their output with the reference alignments or by using either of the evaluation platforms. They can tune their systems with respect to the non-blind evaluation as long as they respect the rules of the OAEI. Participants were required to register their systems and make a preliminary evaluation by July 31^{st}. The execution phase was terminated on September 9^{th}, 2018, at which date participants had to submit the (near) final versions of their systems (SEALS-wrapped and/or HOBBIT-wrapped).

During the **evaluation phase**, systems were evaluated by all track organizers. In case minor problems were found during the initial stages of this phase, they were reported to developers, who were given the opportunity to fix and resubmit their systems. Initial results were provided directly to the participants, whereas final results for most tracks were published on the respective pages of the OAEI website by October 8^{th}.

---

[4] `https://project-hobbit.eu/outcomes/hobbit-platform/`

3 Tracks and test cases

This year's OAEI campaign consisted of 12 tracks gathering 23 test cases, all of which were based on OWL ontologies. They can be grouped into:

- Schema matching tracks, which have as objective matching ontology classes and/or properties.
- Instance Matching tracks, which have as objective matching ontology instances.
- Instance and Schema Matching tracks, which involve both of the above.
- Complex Matching tracks, which have as objective finding complex correspondences between ontology entities.
- Interactive tracks, which simulate user interaction to enable the benchmarking of interactive matching algorithms.

The tracks are summarized in Table 1.

Table 1. Characteristics of the OAEI tracks.

Track	Test Cases (Tasks)	Relations	Confidence	Evaluation	Languages	Platform
Schema Matching						
Anatomy	1	=	[0 1]	open	EN	SEALS
Biodiversity & Ecology	2	=	[0 1]	open	EN	SEALS
Conference	1 (21)	=, <=	[0 1]	open+blind	EN	SEALS
Disease & Phenotype	2	=, <=	[0 1]	open+blind	EN	SEALS
Large Biomedical ontologies	6	=	[0 1]	open	EN	both
Multifarm	2 (2695)	=	[0 1]	open+blind	AR, CZ, CN, DE, EN, ES, FR, IT, NL, RU, PT	SEALS
Instance Matching						
IIMB	1	=	[0 1]	open+blind	EN	SEALS
Link Discovery	2 (9)	=	[0 1]	open	EN	HOBBIT
SPIMBENCH	2	=	[0 1]	open+blind	EN	HOBBIT
Instance and Schema Matching						
Knowledge Graph	9	=	[0 1]	open	EN	both
Interactive Matching						
Interactive	2 (22)	=, <=	[0 1]	open	EN	SEALS
Complex Matching						
Complex	4	=	[0 1]	open+blind	EN, ES	SEALS

Open evaluation is made with already published reference alignments and blind evaluation is made by organizers, either from reference alignments unknown to the participants or manually.

3.1 Anatomy

The anatomy track comprises a single test case consisting of matching two fragments of biomedical ontologies which describe the human anatomy[5] (3304 classes) and the anatomy of the mouse[6] (2744 classes). The evaluation is based on a manually curated reference alignment. This dataset has been used since 2007 with some improvements over the years [13].

Systems are evaluated with the standard parameters of precision, recall, F-measure. Additionally, recall+ is computed by excluding trivial correspondences (i.e., correspondences that have the same normalized label). Alignments are also checked for coherence using the Pellet reasoner. The evaluation was carried out on a server with a 6 core CPU @ 3.46 GHz with 8GB allocated RAM, using the SEALS client. However, the evaluation parameters were computed *a posteriori*, after removing from the alignments produced by the systems s expressing relations other than equivalence, as well as trivial correspondences in the oboInOwl namespace (e.g., oboInOwl#Synonym = oboInOwl#Synonym). The results obtained with the SEALS client vary in some cases by 0.5% compared to the results presented below.

3.2 Biodiversity and Ecology

The new biodiversity track features two test cases based on highly overlapping ontologies that are particularly useful for biodiversity and ecology research: matching the Environment Ontology (ENVO) to the Semantic Web for Earth and Environment Technology Ontology (SWEET), and matching the Flora Phenotype Ontology (FLOPO) to the Plant Trait Ontology (PTO). The track was motivated by two projects, namely GFBio[7] (The German Federation for Biological Data) and AquaDiva[8], which aim at providing semantically enriched data management solutions for data capture, annotation, indexing and search [32]. Table 2 summarizes the versions and the sizes of the ontologies used in OAEI 2018.

Table 2. Versions and number of classes of the Biodiversity and Ecology track ontologies.

Ontology	Version	Classes
ENVO	2017-08-22	6909
SWEET	2018-03-12	4543
FLOPO	2016-06-03	24199
PTO	2017-09-11	1504

The reference alignments for the two test cases were produced through a hybrid approach that consisted of (1) using established matching systems to produce an au-

---

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

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

[7] www.gfbio.org

[8] www.aquadiva.uni-jena.de

tomated consensus alignment (akin to those used in the Disease and Phenotype track) then (2) manually validating the unique results produced by each system (and adding them to the consensus if deemed correct), and finally (3) adding manually generated correspondences. The matching systems used were the OAEI 2017 versions of AML, LogMap, LogMapBio, LogMapLite, LYAM, POMap, and YAMBio, in addition to the alignments from BioPortal [38].

The evaluation was carried out on a Windows 10 (64-bit) desktop with an Intel Core i5-7500 CPU @ 3.40GHz x 4 with 15.7 Gb RAM allocated, using the SEALS client. Systems were evaluated using the standard metrics.

3.3 Conference

The conference track features a single test case that is a suite of 21 matching tasks corresponding to the pairwise combination of 7 moderately expressive ontologies describing the domain of organizing conferences. The dataset and its usage are described in [47].

The track uses several reference alignments for evaluation: the old (and not fully complete) manually curated open reference alignment, *ra1*; an extended, also manually curated version of this alignment, *ra2*; a version of the latter corrected to resolve violations of conservativity, *rar2*; and an uncertain version of *ra1* produced through crowd-sourcing, where the score of each correspondences is the fraction of people in the evaluation group that agree with the correspondence. The latter reference was used in two evaluation modalities: *discrete* and *continuous* evaluation. In the former, correspondences in the uncertain reference alignment with a score of at least 0.5 are treated as correct whereas those with lower score are treated as incorrect, and standard evaluation parameters are used to evaluated systems. In the latter, weighted precision, recall and F-measure values are computed by taking into consideration the actual scores of the uncertain reference, as well as the scores generated by the matching system. For the sharp reference alignments (*ra1*, *ra2* and *rar2*), the evaluation is based on the standard parameters, as well the $F_{0.5}$-measure and F_2-measure and on conservativity and consistency violations. Whereas F_1 is the harmonic mean of precision and recall where both receive equal weight, F_2 gives higher weight to recall than precision and $F_{0.5}$ gives higher weight to precision higher than recall.

Two baseline matchers are use to benchmark the systems: edna string edit distance matcher; and StringEquiv string equivalence matcher as in the anatomy test case.

The evaluation was carried out on a Windows 10 (64-bit) desktop with an Intel Core i7–8550U (1,8 GHz, TB 4 GHz) x 4 with 16 GB RAM allocated using the SEALS client. Systems were evaluated using the standard metrics.

3.4 Disease and Phenotype

The Disease and Phenotype is organized by the Pistoia Alliance Ontologies Mapping project team[9]. It comprises 2 test cases that involve 4 biomedical ontologies covering the disease and phenotype domains: Human Phenotype Ontology (HP) versus

---

[9] `http://www.pistoiaalliance.org/projects/ontologies-mapping/`

Mammalian Phenotype Ontology (MP) and Human Disease Ontology (DOID) versus Orphanet and Rare Diseases Ontology (ORDO). Currently, correspondences between these ontologies are mostly curated by bioinformatics and disease experts who would benefit from automation of their workflows supported by implementation of ontology matching algorithms. More details about the Pistoia Alliance Ontologies Mapping project and the OAEI evaluation are available in [23]. Table 3.4 summarizes the versions of the ontologies used in OAEI 2018.

Table 3. Disease and Phenotype ontology versions and sources.

Ontology	Version	Source
HP	2017-06-30	OBO Foundry
MP	2017-06-29	OBO Foundry
DOID	2017-06-13	OBO Foundry
ORDO	v2.4	ORPHADATA

The reference alignments used in this track are silver standard consensus alignments automatically built by merging/voting the outputs of the participating systems in 2016, 2017 and 2018 (with vote=3). Note that systems participating with different variants and in different years only contributed once in the voting, that is, the voting was done by family of systems/variants rather than by individual systems. The HP-MP silver standard thus produced contains 2232 correspondences, whereas the DOID-ORDO one contains 2808 correspondences.

Systems were evaluated using the standard parameters as well as the number of unsatisfiable classes computed using the OWL 2 reasoner HermiT [36]. The evaluation was carried out in a Ubuntu 18 Laptop with an Intel Core i9-8950HK CPU @ 2.90GHz x 12 and allocating 25 Gb RAM.

3.5 Large Biomedical Ontologies

The large biomedical ontologies (largebio) track aims at finding alignments between the large and semantically rich biomedical ontologies FMA, SNOMED-CT, and NCI, which contain 78,989, 306,591 and 66,724 classes, respectively. The track consists of six test cases corresponding to three matching problems (FMA-NCI, FMA-SNOMED and SNOMED-NCI) in two modalities: small overlapping fragments and whole ontologies (FMA and NCI) or large fragments (SNOMED-CT).

The reference alignments used in this track are derived directly from the UMLS Metathesaurus [5] as detailed in [29], then automatically repaired to ensure logical coherence. However, rather than use a standard repair procedure of removing problem causing correspondences, we set the relation of such correspondences to *"?"* (unknown). These "?" correspondences are neither considered positive nor negative when evaluating matching systems, but are simply 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 avoid any bias, correspondences were considered problem causing if they were selected for removal by any of the three established repair algorithms: Alcomo [34], LogMap [28], or AML [39]. The reference alignments are summarized in Table 4.

Table 4. Number of correspondences in the reference alignments of the large biomedical ontologies tasks.

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

The evaluation was carried out in a Ubuntu 18 Laptop with an Intel Core i9-8950HK CPU @ 2.90GHz x 12 and allocating 25 Gb of RAM. Evaluation was based on the standard parameters (modified to account for the "?" relations) as well as the number of unsatisfiable classes and the ratio of unsatisfiable classes with respect to the size of the union of the input ontologies. Unsatisfiable classes were computed using the OWL 2 reasoner HermiT [36], or, in the cases in which HermiT could not cope with the input ontologies and the alignments (in less than 2 hours) a lower bound on the number of unsatisfiable classes (indicated by $\geq$) was computed using the OWL 2 EL reasoner ELK [33].

3.6 Multifarm

The multifarm track [35] aims at evaluating the ability of matching systems to deal with ontologies in different natural languages. This dataset results from the translation of 7 ontologies from the conference track (cmt, conference, confOf, iasted, sigkdd, ekaw and edas) into 10 languages: Arabic (ar), Chinese (cn), Czech (cz), Dutch (nl), French (fr), German (de), Italian (it), Portuguese (pt), Russian (ru), and Spanish (es). The dataset is composed of 55 pairs of languages, with 49 matching tasks for each of them, taking into account the alignment direction (e.g. cmt$_{en}$ →edas$_{de}$ and cmt$_{de}$ →edas$_{en}$ are distinct matching tasks). While part of the dataset is openly available, all matching tasks involving the *edas* and *ekaw* ontologies (resulting in 55×24 matching tasks) are used for blind evaluation.

We consider two test cases: i) those tasks where two different ontologies (cmt→edas, for instance) have been translated into two different languages; and ii) those tasks where the same ontology (cmt→cmt) has been translated into two different languages. For the tasks of type ii), good results are not only related to the use of specific techniques for dealing with cross-lingual ontologies, but also on the ability to exploit the identical structure of the ontologies.

The reference alignments used in this track derive directly from the manually curated Conference *ra1* reference alignments. Systems are evaluated using the standard parameters. The evaluation was carried out on a Ubuntu 16.04 machine configured with 16GB of RAM running under a i7-4790K CPU 4.00GHz x 8 processors, using the SEALS client.

3.7 IIMB

The new IIMB (ISLab Instance Matching Benchmark) track features a single test case consisting of 80 instance matching tasks, in which the goal is to match an original OWL Abox to an automatically transformed version of this Abox using the SWING (Semantic Web INstance Generation) framework [22]. SWING consists of a pool of transformation techniques organized as follows:

- *Data value transformations* (DVL) are based on changes of cardinality and content of property values belonging to instance descriptions (e.g. value deletion, value modification through random character insertion/substitution).
- *Data structure transformations* (DST) are based on changes of property names and structure within an instance description (e.g. string value splitting, property name modification).
- *Data semantics transformations* (DSS) are based on changes of class/type properties belonging to instance descriptions (e.g. property type deletion/modification).

The IIMB dataset has been generated by relying on a seed of linked-data instances I' extracted from the web. A set of manipulated instances I'' has been created from I' and inserted in IIMB by applying a combination of SWING transformation techniques according to the following schema:

- Tasks ID 001-020: DVL transformations
- Tasks ID 021-040: DST transformations
- Tasks ID 041-060: DSS transformations
- Tasks ID 061-080: DVL, DST, and DSS transformations

Within a group of tasks, the complexity of applied transformations increases with the task ID. In each task, the reference alignment corresponds to the correspondence-set generated by SWING between the instances of the original and transformed Abox.

The evaluation has been performed on an Intel Xeon E5/Core i7 server with 16GB RAM, the Ubuntu operating systems equipped with the SEALS client.

3.8 Link Discovery

The Link Discovery track features two test cases, Linking and Spatial, that deal with *link discovery* for spatial data represented as *trajectories* i.e., sequences of longitude, latitude pairs. The track is based on two datasets generated from TomTom[10] and Spaten [10].

The **Linking** test case aims at testing the performance of instance matching tools that implement mostly string-based approaches for identifying matching entities. It can be used not only by instance matching tools, but also by SPARQL engines that deal with query answering over geospatial data. The test case was based on SPIMBENCH [40], but since the ontologies used to represent trajectories are fairly simple and do not consider complex RDF or OWL schema constructs already supported by SPIMBENCH, only a subset of the transformations implemented by SPIMBENCH was used.

---

[10] `https://www.tomtom.com/en_gr/`

The transformations implemented in the test case were (I) string-based with different (a) levels, (b) types of spatial object representations and (c) types of date representations, and (II) schema-based, i.e., addition and deletion of ontology (schema) properties. These transformations were implemented in the TomTom dataset. In a nutshell, instance matching systems are expected to determine whether two traces with their points annotated with place names designate the same trajectory. In order to evaluate the systems we built a ground truth containing the set of expected links where an instance s_1 in the source dataset is associated with an instance t_1 in the target dataset that has been generated as a modified description of s_1.

The *Spatial* test case aims at testing the performance of systems that deal with topological relations proposed in the state of the art DE-9IM (Dimensionally Extended nine-Intersection Model) model [44]. The benchmark generator behind this test case implements all topological relations of DE-9IM between trajectories in the two dimensional space. To the best of our knowledge such a generic benchmark, that takes as input trajectories and checks the performance of linking systems for spatial data does not exist. For the design, we focused on (a) on the correct implementation of all the topological relations of the DE-9IM topological model and (b) on producing large datasets large enough to stress the systems under test. The supported relations are: *E*quals, Disjoint, Touches, Contains/Within, Covers/CoveredBy, Intersects, Crosses, Overlaps. The test case comprises tasks for all the DE-9IM relations and for LineString/LineString and LineString/Polygon cases, for both TomTom and Spaten datasets, ranging from 200 to 2K instances. We did not exceed 64 KB per instance due to a limitation of the Silk system[11], in order to enable a fair comparison of the systems participating in this track.

The evaluation for both test cases was carried out using the HOBBIT platform.

3.9 SPIMBENCH

The SPIMBENCH track consists of matching instances that are found to refer to the same real-world entity corresponding to a creative work (that can be a news item, blog post or programme). The datasets were generated and transformed using SPIMBENCH [40] by altering a set of original linked data through value-based, structure-based, and semantics-aware transformations (simple combination of transformations). They share almost the same ontology (with some differences in property level, due to the structure-based transformations), which describes instances using 22 classes, 31 Data Properties, and 85 Object Properties. Participants are requested to produce a set of correspondences between the pairs of matching instances from the source and target datasets that are found to refer to the same real-world entity. An instance in the source dataset can have none or one matching counterparts in the target dataset. The SPIMBENCH task is composed of two datasets[12] with different scales (i.e., number of instances to match):

- Sandbox (380 INSTANCES, 10000 TRIPLES). It contains two datasets called source (Tbox1) and target (Tbox2) as well as the set of expected correspondences (i.e., reference alignment).

---

[11] `https://github.com/silk-framework/silk/issues/57`

[12] Although the files are called Tbox1 and Tbox2, they actually contain a Tbox and an Abox.

- Mainbox (1800 CWs, 50000 TRIPLES). It contains two datasets called source (Tbox1) and target (Tbox2). This test case is blind, meaning that the reference alignment is not given to the participants. In both datasets, the goal is to discover the correspondences among the instances in the source dataset (Tbox1) and the instances in the target dataset (Tbox2).

The evaluation was carried out using the HOBBIT platform.

3.10 Knowledge Graph

The new Knowledge Graph track consists of nine isolated graphs generated by running the DBpedia extraction framework on nine different Wikis from the Fandom Wiki hosting platform[13] in the course of the DBkWik project [24, 25]. These knowledge graphs cover three different topics, with three knowledge graphs per topic, so the track consists of nine test cases, corresponding to the pairwise combination of the knowledge graphs in each topic. The goal of each test case is to match both the instances and the schema simultaneously. The datasets are summarized in Table 5

Table 5. Characteristics of the Knowledge Graphs in the KG track, and the sources they were created from.

Source	Hub	Topic	#Instances	#Properties	#Classes
RuneScape Wiki	Games	Gaming	200,605	1,998	106
Old School RuneScape Wiki	Games	Gaming	38,563	488	53
DarkScape Wiki	Games	Gaming	19,623	686	65
Marvel Database	Comics	Comics	56,464	99	2
Hey Kids Comics Wiki	Comics	Entertainment	158,234	1,925	181
DC Database	Comics	Lifestyle	128,495	177	5
Memory Alpha	TV	Entertainment	63,240	326	0
Star Trek Expanded Universe	TV	Entertainment	17,659	201	3
Memory Beta	Books	Entertainment	63,223	413	11

The evaluation was based on a gold standard[14] of correspondences both on the schema and the instance level. While the schema level correspondences were created by experts, the instance correspondences were crowd sourced using Amazon MTurk. Since we do not have a correspondence for each instance, class, and property in the graphs, this gold standard is only a *partial gold standard*.

The evaluation was executed on a virtual machine (VM) with 32GB of RAM and 16 vCPUs (2.4 GHz), with Debian 9 operating system and Openjdk version 1.8.0_181, using the SEALS client. It was not executed on the HOBBIT platform because few systems registered in HOBBIT for this task and all of them also had a SEALS counterpart.

We used the `-o` option in SEALS (version 7.0.5) to provide the two knowledge graphs which should be matched. We used local files rather than HTTP URLs to circumvent the overhead of downloading the knowledge graphs. We could not use the

---

[13] `https://www.wikia.com/`

[14] `http://dbkwik.webdatacommons.org`

”-x” option of SEALS because we had to modify the evaluation routine for two reasons. First, we wanted to differentiate between results for class, property, and instance correspondences, and second, we had to change the evaluation to deal with the partial nature of our gold standard.

The alignments were evaluated based on precision, recall, and f-measure for classes, properties, and instances (each in isolation). Our partial gold standard contained 1:1 correspondences, as well as *negative* correspondences, i.e., correspondences stating that a resource `A` in one knowledge graph has *no* correspondence in the second knowledge graph. This allows to increase the count of false positives if the matcher nevertheless finds a correspondence (i.e., maps `A` to a resource in the other knowledge graph). We further assume that in each knowledge graph, only one representation of the concept exists. This means that if we have a correspondence in our gold standard, we count a correspondence to a different concept as a false positive. The count of false negatives is only increased if we have a 1:1 correspondence and it is not found by a matcher. The whole source code for generating the evaluation results is also available[15].

As a benchmark, we employed a simple string matching approach with some out of the box text preprocessing to generate a baseline. The source code for this approach is publicly available[16].

3.11 Interactive Matching

The interactive matching track aims to assess the performance of semi-automated matching systems by simulating user interaction [37, 12]. The evaluation thus focuses on how interaction with the user improves the matching results. Currently, this track does not evaluate the user experience or the user interfaces of the systems [26, 12].

The interactive matching track is based on the datasets from the Anatomy and Conference tracks, which have been previously described. It relies on the SEALS client’s *Oracle* class to simulate user interactions. An interactive matching system can present a collection of correspondences simultaneously to the oracle, which will tell the system whether that correspondence is correct or not. If a system presents up to three correspondences together and each correspondence presented has a mapped entity (i.e., class or property) in common with at least one other correspondence presented, the oracle counts this as a single interaction, under the rationale that this corresponds to a scenario where a user is asked to choose between conflicting candidate correspondences. To simulate the possibility of user errors, the oracle can be set to reply with a given error probability (randomly, from a uniform distribution). We evaluated systems with four different error rates: 0.0 (perfect user), 0.1, 0.2, and 0.3.

In addition to the standard evaluation parameters, we also compute the number of requests made by the system, the total number of distinct correspondences asked, the number of positive and negative answers from the oracle, the performance of the system according to the oracle (to assess the impact of the oracle errors on the system) and finally, the performance of the oracle itself (to assess how erroneous it was).

---

[15] `http://oaei.ontologymatching.org/2018/results/knowledgegraph/kg_track_eval.zip`

[16] `http://oaei.ontologymatching.org/2018/results/knowledgegraph/string_baseline_kg-source.zip`

The evaluation was carried out on a server with 3.46 GHz (6 cores) and 8GB RAM allocated to the matching systems. Each system was run ten times and the final result of a system for each error rate represents the average of these runs. For the Conference dataset with the ra1 alignment, precision and recall correspond to the micro-average over all ontology pairs, whereas the number of interactions is the total number of interactions for all the pairs.

3.12 Complex Matching

The complex matching track is meant to evaluate the matchers based on their ability to generate complex alignments. A complex alignment is composed of complex correspondences typically involving more than two ontology entities, such as o_1*:AcceptedPaper* $\equiv$ o_2*:Paper* $\sqcap$ o_2*:hasDecision.*o_2*:Acceptance*. Four datasets with their own evaluation process have been proposed [46].

The **complex conference** dataset is composed of three ontologies: cmt, conference and ekaw from the conference dataset. The reference alignment was created as a consensus between experts. In the evaluation process, the matchers can take the simple reference alignment *ra1* as input. The precision and recall measures are manually calculated over the complex equivalence correspondences only.

The **Hydrography** dataset consists of matching four different source ontologies (hydro3, hydrOntology-translated, hydrOntology-native, and cree) to a single target ontology (SWO). The evaluation process is based on three subtasks: given an entity from the source ontology, identify all related entities in the source and target ontology; given an entity in the source ontology and the set of related entities, identify the logical relation that holds between them; identify the full complex correspondences. The first subtask was evaluated based on precision and recall and the latter two were evaluated using semantic precision and recall.

The **GeoLink** dataset derives from the homonymous project, funded under the U.S. National Science Foundation's EarthCube initiative. It is composed of two ontologies: the GeoLink Base Ontology (GBO) and the GeoLink Modular Ontology (GMO). The GeoLink project is a real-world use case of ontologies, and instance data is available. The alignment between the two ontologies was developed in consultation with domain experts from several geoscience research institutions. More detailed information on this benchmark can be found in [48]. Evaluation was done in the same way as with the Hydrography dataset. The evaluation platform was a MacBook Pro with a 2.6 GHz Intel Core i5 processor and 16 GB of 1600 MHz DDR3 RAM running macOS Mojave version 10.14.2.

The **Taxon** dataset is composed of four knowledge bases containing knowledge about plant taxonomy: AgronomicTaxon, AGROVOC, TAXREF-LD and DBpedia. The evaluation is two-fold: first, the precision of the output alignment is manually assessed; then, a set of source queries are rewritten using the output alignment. The rewritten target query is then manually classified as correct or incorrect. A source query is considered successfully rewritten if at least one of the target queries is semantically equivalent to it. The proportion of source queries successfully rewritten is then calculated (QWR in the results table). The evaluation over this dataset is open to all matching systems (simple or complex) but some queries can not be rewritten without complex correspondences. The evaluation was performed with an Ubuntu 16.04 machine configured with 16GB of RAM running under a i7-4790K CPU 4.00GHz x 8 processors.

4 Results and Discussion

4.1 Participation

Following an initial period of growth, the number of OAEI participants has remained approximately constant since 2012, at slightly over 20. This year we observed a slight decrease to 19 participating systems. Table 6 lists the participants and the tracks in which they competed. Some matching systems participated with different variants (AML, LogMap) whereas others were evaluated with different configurations, as requested by developers (see test case sections for details).

Table 6. Participants and the status of their submissions.

System	ALIN	ALOD2vec	AML	AMLC	CANARD	DOME	EVOCROS	FCAMapX	Holontology	KEPLER	Lily	LogMap	LogMap-Bio	LogMapLt	POMAP++	RADON	SANOM	Silk	XMap	Total=19
Confidence	-	-	✓	✓	✓	✓	✓	✓	-	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	14
Anatomy	●	●	●	○	○	●	○	●	●	●	●	●	●	●	●	○	●	○	●	14
Biodiversity & Ecology	○	○	●	○	○	○	○	○	○	◐	●	●	●	●	●	○	○	○	●	8
Conference	●	●	●	○	○	●	○	●	●	●	●	●	○	●	○	○	●	○	●	12
Disease & Phenotype	○	○	●	○	○	●	○	○	○	◐	●	●	●	●	●	○	○	○	●	9
Large Biomedical Ont.	○	◐	●	○	○	●	○	●	○	◐	○	●	●	●	◐	○	○	○	●	10
Multifarm	○	○	●	○	○	◐	◐	○	○	◐	○	●	○	○	○	○	○	○	●	6
IIMB	○	○	●	○	○	○	○	○	○	○	○	●	○	○	○	○	○	○	○	2
Link Discovery	○	○	●	○	○	○	○	○	○	○	○	○	○	○	○	◐	○	◐	○	3
SPIMBENCH	○	○	●	○	○	○	○	○	○	○	●	●	○	○	○	○	○	○	○	3
Knowledge Graph	○	○	◐	○	○	●	○	○	●	○	○	◐	●	●	●	○	○	○	○	7
Interactive Matching	●	○	●	○	○	○	○	○	○	○	○	●	○	○	○	○	○	○	●	4
Complex Matching	○	◐	◐	◐	◐	◐	○	◐	◐	◐	○	◐	◐	◐	◐	○	○	○	◐	13
total	3	4	12	1	1	7	1	4	4	7	5	11	6	7	6	1	2	1	8	65

Confidence pertains to the confidence scores returned by the system, with ✓ indicating that they are non-boolean; ○ indicates that the system did not participate in the track; ● indicates that it participated fully in the track; and ◐ indicates that it participated in or completed only part of the tasks of the track.

A number of participating systems use external sources of background knowledge, which are especially critical in matching ontologies in the biomedical domain. LogMap-Bio uses BioPortal as mediating ontology provider, that is, it retrieves from BioPortal the most suitable top-10 ontologies for each matching task. LogMap uses normalizations and spelling variants from the general (biomedical) purpose SPECIALIST Lexicon. AML has three sources of background knowledge which can be used as mediators between the input ontologies: the Uber Anatomy Ontology (Uberon), the Human Disease Ontology (DOID) and the Medical Subject Headings (MeSH). XMAP and Lily use a dictionary of synonyms (pre)extracted from the UMLS Metathesaurus. In addition Lily also uses a dictionary of synonyms (pre)extracted from BioPortal.

4.2 Anatomy

The results for the Anatomy track are shown in Table 7.

Table 7. Anatomy results, ordered by F-measure. Runtime is measured in seconds; "size" is the number of correspondences in the generated alignment.

System	Runtime	Size	Precision	F-measure	Recall	Recall+	Coherent
AML	42	1493	0.95	0.943	0.936	0.832	√
LogMapBio	808	1550	0.888	0.898	0.908	0.756	√
POMAP++	210	1446	0.919	0.897	0.877	0.695	-
XMap	37	1413	0.929	0.896	0.865	0.647	√
LogMap	23	1397	0.918	0.88	0.846	0.593	√
SANOM	487	1450	0.888	0.865	0.844	0.632	-
FCAMapX	118	1274	0.941	0.859	0.791	0.455	-
KEPLER	244	1173	0.958	0.836	0.741	0.316	-
Lily	278	1382	0.872	0.832	0.795	0.518	-
LogMapLite	18	1147	0.962	0.828	0.728	0.288	-
ALOD2Vec	75	987	0.996	0.785	0.648	0.086	-
StringEquiv	-	946	0.997	0.766	0.622	0.000	-
DOME	22	935	0.997	0.761	0.615	0.009	-
ALIN	271	928	0.998	0.758	0.611	0.0	√
Holontology	265	456	0.976	0.451	0.294	0.005	-

Of the 14 systems participating in the Anatomy track, 11 achieved an F-measure higher than the StringEquiv baseline. Three systems were first time participants (ALOD2Vec, DOME, and Holontology) and showed modest results in terms of both F-measure and recall+, with only ALOD2Vec ranking above the baseline. Among the five systems that participated for the second time, SANOM shows increases in both F-measure (from 0.828 to 0.865) and recall+ (from 0.419 to 0.632), KEPLER and Lily have the same performance as last year, and both POMAP++ (POMap in 2017) and FCAMapX (FCA_Map in 2016) have decreases in F-measure and recall+. Long-term systems showed few changes in comparison with previous years with respect to alignment quality (precision, recall, F-measure, and recall+), size or run time. The exceptions were LogMapBio which increased in both recall+ (from 0.733 to 0.756) and alignment size (by 16 correspondences) since last year, and ALIN that had a substantial increase of 412 correspondences since last year.

In terms of run time, 6 out of 14 systems computed an alignment in less than 100 seconds, a ratio which is similar to 2017 (5 out of 11). LogMapLite remains the system with the shortest runtime. Regarding quality, AML remains the system with the highest F-measure (0.943) and recall+ (0.832), but 4 other systems obtained an F-measure above 0.88 (LogMapBio, POMap++, XMap, and LogMap) which is at least as good as the best systems in OAEI 2007-2010. Like in previous years, there is no significant correlation between the quality of the generated alignment and the run time. Five systems produced coherent alignments, which is the same as last year.

4.3 Biodiversity and Ecology

Of the 8 participants registered for this track, 7 systems (AML, LogMap, LogMapBio, LogMapLt, Lily, XMap and POMap) managed to generate a meaningful output in 4 hours, and only KEPLER did not. Table 8 shows the results for the FLOPO-PTO and ENVO-SWEET tasks.

Table 8. Results for the Biodiversity & Ecology track, ordered by F-measure.

System	Size	Precision	F-measure	Recall
FLOPO-PTO task				
AML	233	0.88	0.86	0.84
LogMap	235	0.817	0.802	0.787
LogMapBio	239	0.803	0.795	0.787
XMap	153	0.987	0.761	0.619
LogMapLite	151	0.987	0.755	0.611
POMap	261	0.663	0.685	0.709
LiLy	176	0.813	0.681	0.586
ENVO-SWEET task				
AML	791	0,776	0,844	0,926
LogMap	583	0,839	0,785	0,738
POMap	583	0,839	0,785	0,738
XMap	547	0,868	0,785	0,716
LogMapBio	572	0,839	0,777	0,724
LogMapLite	740	0,732	0,772	0,817
LiLy	491	0,866	0,737	0,641

Regarding the FLOPO-PTO task, the top 3 ranked systems in terms of F-measure are AML, LogMap and LogMapBio, with curiously a similar number of generated correspondences among them. Among these, AML achieved the highest F-measure (0.86) and a well-balanced result, with over 80% recall and a still quite high precision.

Regarding the ENVO-SWEET task, AML ranked first in terms of F-measure, followed by a three-way tie between LogMap, POMAP and XMap. AML had a less balanced result in this test case, with a very high recall and significant larger alignment than the other top systems, but a comparably lower precision. LogMap and POMap produced alignments of exactly equal size and quality, whereas XMap had the highest precision overall, but a lower recall than these.

Overall, in this first evaluation, the results obtained from participating systems are quite promising, as all systems achieved more than 0.68 in term of F-measure. We should note that most of the participating systems, and all of the most successful ones use external resources as background knowledge.

4.4 Conference

The conference evaluation results using the sharp reference alignment *rar2* are shown in Table 9. For the sake of brevity, only results with this reference alignment and considering both classes and properties are shown. For more detailed evaluation results, please check conference track's web page.

With regard to the two baselines we can group the twelve participants into four groups: six matching systems outperformed both baselines (SANOM, AML, LogMap, XMap, FCAMapX and DOME); three performed the same as the edna baseline (ALIN, LogMapLt and Holontology); two performed slightly worse than this baseline (KEPLER and ALOD2Vec); and Lily performed worse than both baselines. Note that two systems (ALIN and Lily) do not match properties at all which naturally has a negative effect on their overall performance.

The performance of all matching systems regarding their precision, recall and F_1-measure is plotted in Figure 1. Systems are represented as squares or triangles, whereas the baselines are represented as circles.

With respect to logical coherence [42, 43], only three tools (ALIN, AML and LogMap) have no consistency principle violation (in comparison to five tools last year and seven tools two years ago). This year all tools have some conservativity principle violations (in comparison to one tool having no conservativity principle violation last year). We should note that these conservativity principle violations can be "false positives" since the entailment in the aligned ontology can be correct although it was not derivable in the single input ontologies.

The Conference evaluation results using the uncertain reference alignments are presented in Table 10.

Among the twelve participating alignment systems, six use 1.0 as the confidence value for all matches they identify (ALIN, ALOD2Vec, DOME, FCAMapX, Holontology, LogMapLt), whereas the remaining six have a wide range of confidence values (AML, KEPLER, Lily, LogMap, SANOM and XMap).

When comparing the performance of the matchers on the uncertain reference alignments versus that on the sharp version (with the corresponding *ra1*), we see that in the discrete case all matchers except Lily performed the same or better in terms of F-measure (Lily's F-measure dropped by 0.01). The changes in F-measure ranged from -1 to 15 percent over the sharp reference alignment. This was predominantly driven by increased recall, which is a result of the presence of fewer 'controversial' matches in the uncertain version of the reference alignment.

The performance of the matchers with confidence values always 1.0 is very similar regardless of whether a discrete or continuous evaluation methodology is used, because many of their correspondences are ones that the experts had high agreement

Table 9. The highest average $F_{[0.5|1|2]}$-measure and their corresponding precision and recall for each matcher with its F_1-optimal threshold (ordered by F_1-measure). Inc.Align. means number of incoherent alignments. Conser.V. means total number of all conservativity principle violations. Consist.V. means total number of all consistency principle violations.

System	Prec.	$F_{0.5}$-m.	F_1-m.	F_2-m.	Rec.	Inc.Align.	Conser.V.	Consist.V.
SANOM	0.72	0.71	0.7	0.69	0.68	9	103	92
AML	0.78	0.74	0.69	0.65	0.62	0	39	0
LogMap	0.77	0.72	0.66	0.6	0.57	0	25	0
XMap	0.76	0.7	0.62	0.56	0.52	4	53	14
FCAMapX	0.64	0.62	0.59	0.56	0.54	11	124	273
DOME	0.74	0.66	0.57	0.5	0.46	3	106	10
edna	*0.74*	*0.66*	*0.56*	*0.49*	*0.45*			
ALIN	0.82	0.69	0.56	0.48	0.43	0	2	0
Holontology	0.73	0.65	0.56	0.49	0.45	3	66	10
LogMapLt	0.68	0.62	0.56	0.5	0.47	5	96	25
ALOD2Vec	0.67	0.62	0.55	0.5	0.47	6	124	27
KEPLER	0.67	0.61	0.55	0.49	0.46	12	123	159
StringEquiv	*0.76*	*0.65*	*0.53*	*0.45*	*0.41*			
Lily	0.54	0.53	0.52	0.51	0.5	9	140	124

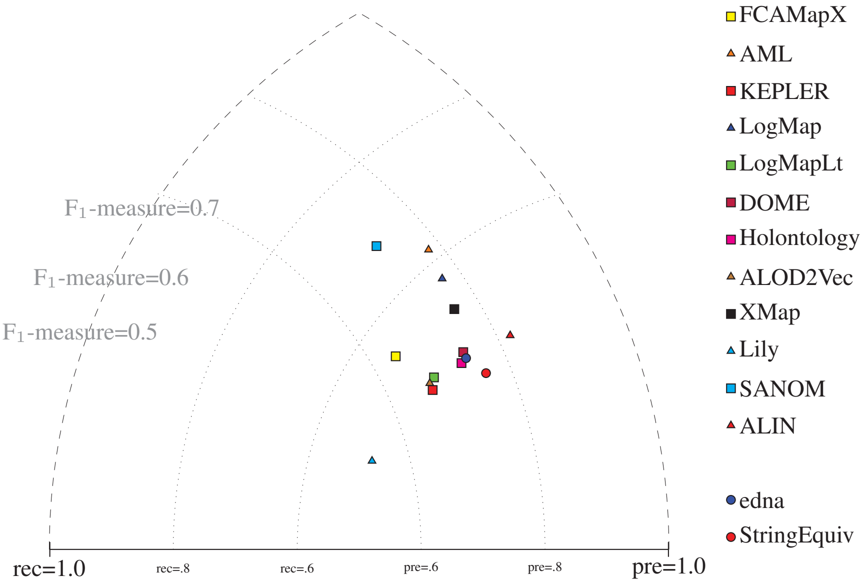


Fig. 1. Precision/recall triangular graph for the conference test case. Dotted lines depict level of precision/recall while values of F_1-measure are depicted by areas bordered by corresponding lines F_1-measure=0.[5|6|7].

Table 10. F-measure, precision, and recall of matchers when evaluated using the sharp (*ra1*), discrete uncertain and continuous uncertain metrics. Sorted according to F_1-m. in continuous.

System	Sharp			Discrete			Continuous		
	Prec.	F_1-m.	Rec.	Prec.	F_1-m.	Rec.	Prec.	F_1-m.	Rec.
AML	0.84	0.74	0.66	0.79	0.78	0.77	0.80	0.77	0.74
SANOM	0.79	0.74	0.69	0.71	0.74	0.78	0.65	0.72	0.81
ALIN	0.88	0.60	0.46	0.88	0.69	0.57	0.88	0.70	0.59
XMap	0.81	0.65	0.54	0.66	0.74	0.83	0.74	0.70	0.66
DOME	0.79	0.60	0.48	0.79	0.68	0.60	0.78	0.69	0.62
Holontology	0.78	0.59	0.48	0.78	0.68	0.60	0.78	0.68	0.61
ALOD2Vec	0.71	0.59	0.50	0.71	0.66	0.62	0.71	0.67	0.63
LogMap	0.82	0.69	0.59	0.77	0.73	0.70	0.80	0.67	0.57
LogMapLt	0.73	0.59	0.50	0.73	0.67	0.62	0.72	0.67	0.63
FCAMapX	0.68	0.61	0.56	0.65	0.66	0.67	0.64	0.66	0.68
KEPLER	0.76	0.59	0.48	0.76	0.67	0.60	0.58	0.63	0.68
Lily	0.59	0.56	0.53	0.52	0.55	0.59	0.59	0.32	0.22

about, while the ones they missed were more controversial. AML produces a fairly wide range of confidence values and has the highest F-measure under both the continuous and discrete evaluation methodologies, indicating that this system's confidence evaluation does a good job of reflecting cohesion among experts on this task. Of the remaining systems, four (KEPLER, LogMap, SANOM and XMap) have relatively small drops in F-measure when moving from discrete to continuous evaluation. Lily's performance drops drastically under the continuous evaluation methodology. This is because the matcher assigns low confidence values to some correspondences in which the labels are equivalent strings, which many experts agreed with unless there was a compelling reason not to. This hurts recall, but using a low threshold value in the discrete version of the evaluation metrics 'hides' this problem.

Overall, in comparison with last year, the F-measures of most returning matching systems essentially held constant under both the sharp and uncertain evaluations. The exceptions were ALIN and SANOM, whose performance improved substantially. In fact, the latter improved its performance so much that it became the top system with regard to F-measure according to the sharp evaluation. We can conclude that all matchers perform better on the fuzzy versus sharp version of the benchmark and that the performance of AML against the fuzzy reference alignment rivals that of a human evaluated in the same way.

4.5 Disease and Phenotype Track

In the OAEI 2018 phenotype track 9 systems were able to complete at least one of the tasks with a 6 hours timeout. Tables 11 show the evaluation results in the HP-MP and DOID-ORDO matching tasks, respectively.

Since the consensus reference alignments only allow us to assess how systems perform in comparison with one another, the proposed ranking is only a reference. Note that some of the correspondences in the consensus alignment may be erroneous (false

Table 11. Results for the HP-MP and DOID-ORDO tasks based on the consensus reference alignment.

System	Time (s)	# Corresp.	# Unique	Scores			Incoherence	
				Prec.	F-m.	Rec.	Unsat.	Degree
HP-MP task								
LogMap	31	2,130	1	0.88	0.86	0.84	0	0.0%
LogMapBio	821	2,178	37	0.86	0.85	0.84	0	0.0%
AML	70	2,010	279	0.89	0.84	0.80	0	0.0%
LogMapLt	7	1,370	3	0.99	0.76	0.61	0	0.0%
POMAP++	1,668	1,502	214	0.86	0.69	0.58	0	0.0%
Lily	4,749	2,118	733	0.68	0.66	0.65	0	0.0%
XMap	20	704	2	0.99	0.48	0.31	0	0.0%
DOME	46	689	0	1.00	0.47	0.31	0	0.0%
DOID-ORDO task								
LogMap	25	2,323	0	0.94	0.85	0.78	0	0.0%
LogMapBio	1,891	2,499	91	0.90	0.85	0.80	0	0.0%
POMAP++	2,264	2,563	174	0.87	0.83	0.80	0	0.0%
LogMapLt	7	1,747	16	0.99	0.76	0.62	0	0.0%
XMap	15	1,587	37	0.97	0.70	0.55	0	0.0%
KEPLER	2,746	1,824	158	0.88	0.70	0.57	0	0.0%
Lily	2,847	3,738	1,167	0.59	0.67	0.78	206	1.9%
AML	135	4,749	1,886	0.51	0.65	0.87	0	0.0%
DOME	10	1,232	2	1.00	0.61	0.44	0	0.0%

positives) because all systems that agreed on it could be wrong (e.g., in erroneous correspondences with equivalent labels, which are not that uncommon in biomedical tasks). In addition, the consensus alignments will not be complete, because there are likely to be correct correspondences that no system is able to find, and there are a number of correspondences found by only one system (and therefore not in the consensus alignments) which may be correct. Nevertheless, the results with respect to the consensus alignments do provide some insights into the performance of the systems.

Overall, LogMap is the system that provides the closest set of correspondences to the consensus (not necessarily the best system) in both tasks. It has a small set of unique correspondences as most of its correspondences are also suggested by its variant LogMapBio and vice versa. By contrast, Lily and AML produce the highest number of unique correspondences in HP-MP and DOID-ORDO respectively, and the second-highest inversely. All systems produce coherent alignments except for Lily in the DOID-ORDO task.

4.6 Large Biomedical Ontologies

In the OAEI 2018 Large Biomedical Ontologies track, 10 systems were able to complete at least one of the tasks within a 6 hours timeout. Seven systems were able to complete all six tasks.[17] Since the reference alignments for this track are based on the

---

[17] Check out the supporting scripts to reproduce the evaluation: `https://github.com/ernestojimenezruiz/oaei-evaluation`

Table 12. Results for the whole ontologies matching tasks in the OAEI largebio track.

System	Time (s)	# Corresp.	# Unique	Scores			Incoherence	
				Prec.	F-m.	Rec.	Unsat.	Degree
Whole FMA and NCI ontologies (Task 2)								
AML	55	2,968	311	0.84	0.86	0.87	2	0.014%
LogMap	1,072	2,701	0	0.86	0.83	0.81	2	0.014%
LogMapBio	1,072	2,860	39	0.83	0.83	0.83	2	0.014%
XMap2	65	2,415	52	0.88	0.80	0.74	2	0.014%
FCAMapX	881	3,607	443	0.67	0.74	0.84	8,902	61.8%
LogMapLt	6	3,458	250	0.68	0.74	0.82	5,170	35.9%
DOME	12	2,383	10	0.80	0.73	0.67	596	4.1%
Whole FMA ontology with SNOMED large fragment (Task 4)								
FCAMapX	1,736	7,971	1,258	0.82	0.79	0.76	21,289	57.0%
AML	94	6,571	462	0.88	0.77	0.69	0	0.0%
LogMapBio	1,840	6,471	31	0.83	0.73	0.65	0	0.0%
LogMap	288	6,393	0	0.84	0.73	0.65	0	0.0%
XMap2	299	6,749	1,217	0.72	0.66	0.61	0	0.0%
LogMapLt	9	1,820	56	0.85	0.33	0.21	981	2.6%
DOME	20	1,588	1	0.94	0.33	0.20	951	2.5%
Whole NCI ontology with SNOMED large fragment (Task 6)								
AML	168	13,176	1,230	0.90	0.77	0.67	≥517	≥0.6%
FCAMapX	2,377	15,383	1,670	0.80	0.73	0.68	≥72,859	≥85.5%
LogMapBio	2,942	13,098	231	0.85	0.72	0.63	≥3	≥0.004%
LogMap	475	12,276	0	0.87	0.71	0.60	≥1	≥0.001%
LogMapLt	11	12,864	720	0.80	0.66	0.57	≥74,013	≥86.9%
DOME	24	9,702	42	0.91	0.63	0.49	≥53,574	≥62.9%
XMap2	427	16,271	4,432	0.64	0.61	0.58	≥73,571	≥86.4%

UMLS-Metathesaurus, we disallowed the use of this resource as a source of background knowledge in the matching systems that used it, XMap and Lily. XMap was still able to produce competitive results, while Lily produced an empty set of alignments. The evaluation results for the largest matching tasks are shown in Tables 12.

The top-ranked systems by F-measure were respectively: AML and LogMap in Task 2; FCAMapX and AML in Task 4; and AML and FCAMapX in Task 6.

Interestingly, the use of background knowledge led to an improvement in recall from LogMap-Bio over LogMap in all tasks, but this came at the cost of precision, resulting in the two variants of the system having very similar F-measures.

The effectiveness of all systems decreased from small fragments to whole ontologies tasks.[18] One reason for this is that with larger ontologies there are more plausible correspondence candidates, and thus it is harder to attain both a high precision and a high recall. In fact, this same pattern is observed moving from the FMA-NCI to the FMA-SNOMED to the SNOMED-NCI problem, as the size of the task also increases. Another reason is that the very scale of the problem constrains the matching strategies

---

[18] `http://www.cs.ox.ac.uk/isg/projects/SEALS/oaei/2018/results/`

that systems can employ: AML for example, forgoes its matching algorithms that are computationally more complex when handling very large ontologies, due to efficiency concerns.

The size of the whole ontologies tasks proved a problem for a number of systems, which were unable to complete them within the allotted time: POMAP++, ALOD2Vec and KEPLER.

With respect to alignment coherence, as in previous OAEI editions, only three distinct systems have shown alignment repair facilities: AML, LogMap and its LogMap-Bio variant, and XMap (which reuses the repair techniques from Alcomo [34]). Note that only LogMap and LogMap-Bio are able to reduce to a minimum the number of unsatisfiable classes across all tasks, missing 9 unsatisfiable classes in the worst case (whole FMA-NCI task). XMap seems to deactivate the repair facility for the SNOMED-NCI case.

As the results tables show, even the most precise alignment sets may lead to a huge number 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. We encourage ontology matching system developers to develop their own repair techniques or to use state-of-the-art techniques such as Alcomo [34], the repair module of LogMap (LogMap-Repair) [28] or the repair module of AML [39], which have worked well in practice [30, 21].

4.7 Multifarm

This year, 6 matching systems registered for the MultiFarm track: AML, DOME, EVOCROS, KEPLER, LogMap and XMap. This represents a slight decrease from the last two years, but is within an approximately constant trend (8 in 2017, 7 in 2016, 5 in 2015, 3 in 2014, 7 in 2013, and 7 in 2012). However, a few systems had issues when evaluated: i) KEPLER generated some parsing errors for some pairs; ii) EVOCROS took around 30 minutes to complete a single task (we have hence tested only 50 matching tasks) and generated empty alignments; iii) DOME was not able to generate any alignment; iv) XMap had problems dealing with most pairs involving the ar, ru and cn languages. Please refer to the OAEI papers of the matching systems for a detailed description of the strategies employed by each system, most of which adopt a translation step before the matching itself.

The Multifarm evaluation results based on the blind dataset are presented in Table 13. They have been computed using the Alignment API 4.9 and can slightly differ from those computed with the SEALS client. We do not report the results of non-specific systems here, as we could observe in the last campaigns that they can have intermediate results in the "same ontologies" task (ii) and poor performance in the "different ontologies" task (i).

With respect to run time, we observe large differences between systems due to the high number of matching tasks involved (55 x 24). Note as well that the concurrent access to the SEALS repositories during the evaluation period may have an impact on the time required for completing the tasks.

Table 13. MultiFarm aggregated results per matcher, for each type of matching task – different ontologies (i) and same ontologies (ii).

System	Time	#pairs	Type (i) – 22 tests per pair				Type (ii) – 2 tests per pair			
			Size	Prec.	F-m.	Rec.	Size	Prec.	F-m.	Rec.
AML	26	55	6.87	.72 (.72)	.46 (.46)	.35 (.35)	23.24	.96 (.95)	.27 (.27)	.16 (.16)
KEPLER	900	53	9.74	.40 (.42)	.27 (.28)	.21 (.22)	58.28	.85 (.88)	.49 (.51)	.36 (.37)
LogMap	39	55	6.99	.72 (.72)	.37 (.37)	.25 (.25)	46.80	.95 (.96)	.41 (.42)	.28 (.28)
XMap	22	26	94.72	.02 (.05)	.03 (.07)	.07 (.07)	345.00	.13 (.18)	.14 (.20)	.19 (.19)

Time is measured in minutes (for completing the 55×24 matching tasks); #pairs indicates the number of pairs of languages for which the tool is able to generate (non-empty) alignments; size indicates the average of the number of generated correspondences for the tests where an (non-empty) alignment has been generated. Two kinds of results are reported: those not distinguishing empty and erroneous (or not generated) alignments and those—indicated between parenthesis—considering only non-empty generated alignments for a pair of languages.

In terms of F-measure, AML remains the top performing system in task (i), followed by LogMap and KEPLER. In task (ii), AML has relatively low performance (with a notably low recall) and KEPLER has the highest F-measure, followed by LogMap.

With respect to the pairs of languages for test cases of type (i), for the sake of brevity, we do not present the detailed results. Please refer to the OAEI results web page to view them. The language pairs in which systems perform better in terms of F-measure include: es-it, it-pt and nl-pt (AML); cz-pt and de-pt (KEPLER); en-nl (LogMap); and cz-en (XMap). We note also some patterns behind the worst results obtained by systems: ar-cn for AML, and some pairs involving cn for KEPLER and LogMap)

In terms of performance, the F-measure for blind tests remains relatively stable across campaigns. AML and LogMap keep their positions and have similar F-measure with respect to the previous campaigns, as does XMap. As observed in previous campaigns, systems privilege precision over recall, and the results are expectedly below the ones obtained for the original Conference dataset. Cross-lingual approaches remain mainly based on translation strategies and the combination of other resources (like cross-lingual links in Wikipedia, BabelNet, etc.) while strategies such as machine learning, or indirect alignment composition remain under-exploited.

4.8 IIMB

Only two systems participated in the new IIMB track: AML and LogMap. The obtained results are summarized in Table 14[19].

In the results of both AML and LogMap, we note that high-quality performances are provided on test-cases based on DVL transformations. We note that the evaluation results on this kind of matching issues have been improved in the recent years (for instance, see [3] for a comparison against the 2012 version of the IIMB dataset). As a matter of fact, recognition of similarities across instance descriptions with data-value

---

[19] A detailed report of test-case results is provided on `https://islab.di.unimi.it/im_oaei_2018/`.

Table 14. Summary of the IIMB results.

System	Runtime (s)	Precision	Recall	F-measure
Data Value Transformations				
AML	1828	0.893	0.789	0.828
LogMap	4.2	0.896	0.893	0.889
Data Structure Transformations				
AML	2036	0.419	0.433	0.424
LogMap	5.7	0.934	0.985	0.959
Data Semantics Transformations				
AML	6.2	0.747	0.889	0.796
LogMap	4.6	0.855	0.947	0.893
Mixed Transformations				
AML	2083	0.334	0.294	0.295
LogMap	6.5	0.920	0.758	0.819

heterogeneities represents a sort of consolidated matching capability that can be considered as a standard functionality of the current state-of-the-art tools. We also note that promising results are also provided by both the participating tools on test-cases based on DSS transformations. We argue that such a kind of result is due to the capability of both AML and LogMap to cope with incoherence, thus reducing the number of false-positive results. As a final remark, we observe that recall is usually lower than precision. Maybe, the cause is the non-uniform quality of expected automatically-generated correspondences. Expected correspondences are created by applying a sequence of transformations with different length (i.e., number of transformations) and different degree of complexity (i.e., strength of applied data manipulations). Sometimes, the applied SWING transformations produce correspondences that are more difficult to agree with, rather than to detect. Measuring the quality of automatically-generated alignments as well as pruning of excessively-hard ones from the set of expected results is a challenging issue to consider in future research work (see Section 5).

4.9 Link Discovery

This year the Link Discovery track counted one participant in the Linking test case (AML) and three participants in the Spatial test case: AML, Silk and RADON.

In the Linking test case, AML perfectly captures all the correct links while not producing wrong ones, thus obtaining perfect precision and a recall (1.0) in both the Sandbox and Mainbox datasets. It required 6.8s and 313s, respectively, to complete the two tasks.

We divided the Spatial test cases into four suites. In the first two suites (SLL and LLL), the systems were asked to match LineStrings to LineStrings considering a given relation for 200 and 2K instances for the TomTom and Spaten datasets. In the last two tasks (SLP, LLP), the systems were asked to match LineStrings to Polygons (or Polygons to LineStrings depending on the relation) again for both datasets. Since the precision, recall and f-measure results from all systems were equal to 1.0, we are only presenting results regarding the time performance. The time performance of the matching systems in the SLL, LLL, SLP and LLP suites are shown in Figures 2-3.

In the SLL suite, RADON has the best performance in most cases except for the *Touches* and *Intersects* relations, followed by AML. Silk seems to need the most time, particularly for *Touches* and *Intersects* relations in the TomTom dataset and *Overlaps* in both datasets.

In the LLL suite we have a more clear view of the capabilities of the systems with the increase in the number of instances. In this case, RADON and Silk have similar behavior as in the the small dataset, but it is more clear that the systems need much more time to match instances from the TomTom dataset. RADON has still the best performance in most cases. AML has the next best performance and is able to handle some cases better than other systems (e.g. *Touches* and *Intersects*), however, it also hits the platform time limit in the case of *Disjoint*.

In the SLP suite, in contrast to the first two suites, RADON has the best performance for all relations. AML and Silk have minor time differences and, depending on the case, one is slightly better than the other. All the systems need more time for the TomTom dataset but due to the small size of the instances the time difference is minor.

In the LLP suite, RADON again has the best performance in all cases. AML hits the platform time limit in *Disjoint* relations on both datasets and is better than Silk in most cases except *Contains* and *Within* on the TomTom dataset where it needs an excessive amount of time.

Taking into account the executed test cases we can identify the capabilities of the tested systems as well as suggest some improvements. All the systems participated in most of the test cases, with the exception of Silk which did not participate in the *Covers* and *Covered By* test cases.

RADON was the only system that successfully addressed all the tasks, and had the best performance for the SLP and LLP suites, but it can be improved for the *Touches* and *Intersects* relations for the SLL and LLL suites. AML performs extremely well in most cases, but can be improved in the cases of *Covers/Covered By* and *Contains/Within* when it comes to LineStrings/Polygons Tasks and especially in *Disjoint* relations where it hits the platform time limit. Silk can be improved for the *Touches*, *Intersects* and *Overlaps* relations and for the SLL and LLL tasks and for the *Disjoint* relation in SLP and LLP Tasks.

In general, all systems needed more time to match the TomTom dataset than the Spaten one, due to the smaller number of points per instance in the latter. Comparing the LineString/LineString to the LineString/Polygon Tasks we can say that all the systems needed less time for the first for the *Contains*, *Within*, *Covers* and *Covered by* relations, more time for the *Touches*, *Instersects* and *Crosses* relations, and approximately the same time for the *Disjoint* relation.

4.10 SPIMBENCH

This year, the SPIMBENCH track counted three participants: AML, Lily, and LogMap. The evaluation results of the track are shown in Table 15.

Lily had the best performance overall both in terms of F-measure and in terms of run time. Notably, its run time scaled very well with the increase in the number of

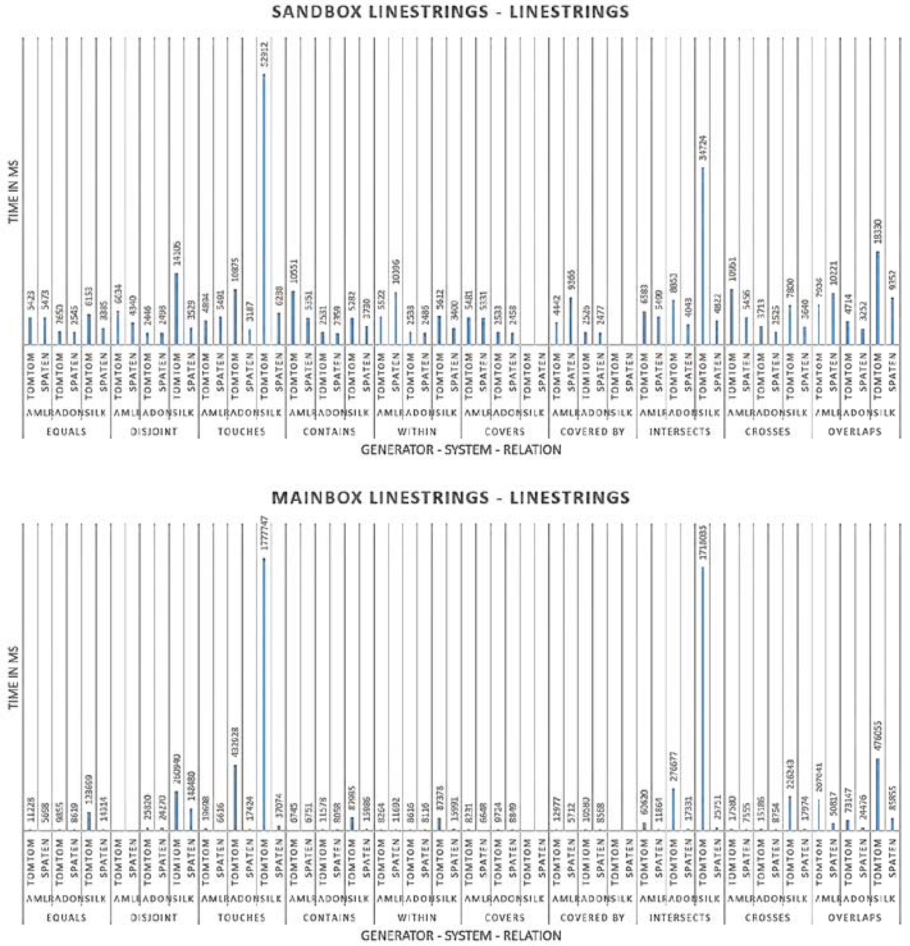


Fig. 2. Time performance for TomTom & Spaten SLL (top) and LLL (bottom) suites for AML (A), Silk (S) and RADON (R).

instances. Both Lily and AML had a higher recall than precision, with the former having full recall. By contrast, LogMap had the highest precision but lowest recall of the three systems. AML and LogMap had a similar run time for the Sandbox task, but the latter scaled better with the increase in the number of instances.

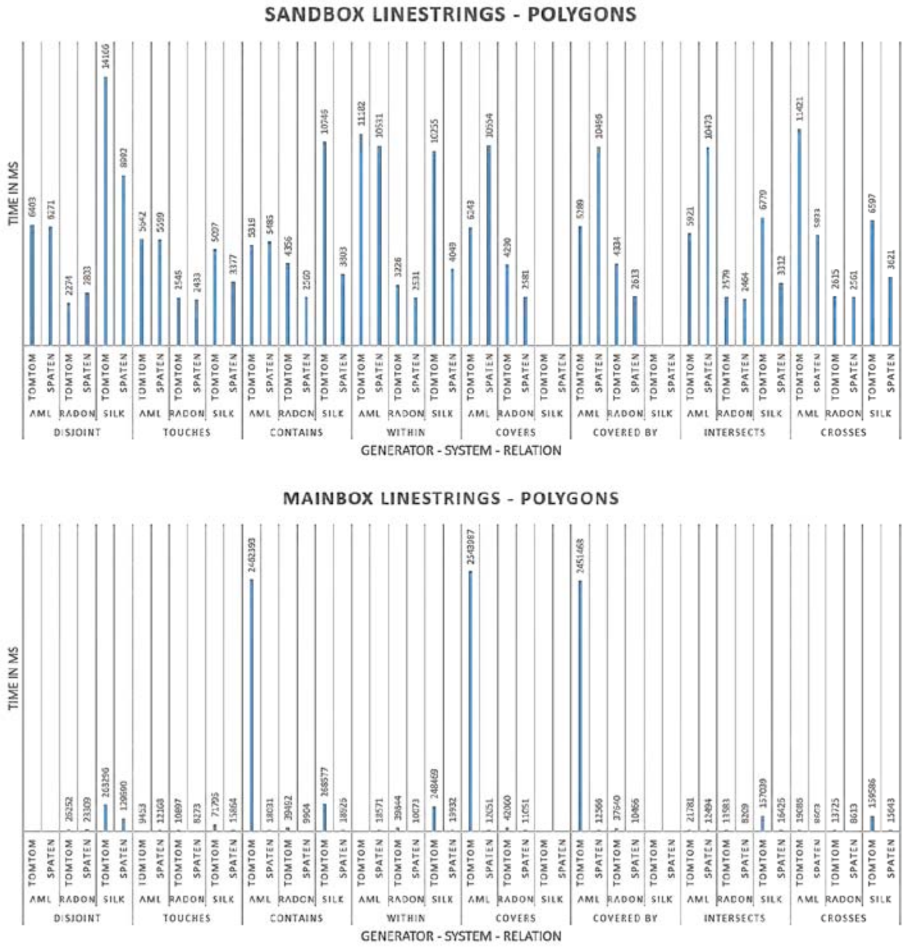


Fig. 3. Time performance for TomTom & Spaten SLP (top) and LLP (bottom) suites for AML (A), Silk (S) and RADON (R).

Table 15. SPIMBENCH track results.

System	Precision	Recall	F-measure	Time (ms)
Sandbox (100 instances)				
AML	0.835	0.896	0.865	6220
Lily	0.849	1.0	0.919	1960
LogMap	0.938	0.763	0.841	5887
Mainbox (5000 instances)				
AML	0.839	0.884	0.860	37190
Lily	0.855	1.0	0.922	3103
LogMap	0.893	0.709	0.791	23494

4.11 Knowledge Graph

We evaluated all SEALS participants in the OAEI (even those not registered for the track) on a very small matching example[20]. This revealed that not all systems were able to cope with the task, and in the end only the following systems were evaluated: AML, POMap++, Hontology, DOME, LogMap (in its KG version), LogMapBio, LogMapLt.

Of these systems, the following were able output results for all nine test cases: POMAP++, Holontology, DOME, LogMapBio and the baseline. AML ran out of time (12 hours) on some tracks, LogMap needed more than the given 32 GB RAM for the bigger knowledge graphs, and LogMapLt created alignment files bigger than 1GB (up to 50 GB in some runs).

Table 16 shows the aggregated results for each system, including the number of tasks in which it was able to generate a non-empty alignment (#tasks) and the average number of generated correspondences in those tasks (size). In addition to the global average precision, F-measure, and recall results, in which tasks where systems produced empty alignments were counted, we also computed F-measure and recall ignoring empty alignments (note that precision is the same) which are shown between parentheses in the table, where applicable.

All systems were able to generate class correspondences, but only the three tasks from the Games topic have enough classes to be meaningfully matched. The baseline has an F-Measure of 0.79 which is surpassed by AML, Holontology, LogMap and LogMapBio (when considering only completed tracks).

DOME was the only system able to produce property correspondences (in addition to the baseline). The remaining systems do not return any property correspondences, probably because all properties are typed as `rdf:Property` and not subdivided into `owl:DatatypeProperty` and `owl:ObjectProperty`. However, this cannot be done easily in a preprocessing step because the usage of the properties is not strict, i.e., some properties are used both with literals and resources as their object. Given that a system that matches only OWL properties of the same type would not be able to handle such cases as this, an improvement of these matching systems would be to include also the ability of correspondence `rdf:Property` in case no more types are defined.

---

[20] `http://oaei.ontologymatching.org/2018/results/knowledgegraph/small_test.zip`

Table 16. Knowledge Graph track results, divided into class, property, instance, and overall correspondences.

System	Time (s)	# tasks	Size	Prec.	F-m.	Rec.
Class performance						
AML	88448	5	11.6	0.85	0.64 (0.87)	0.51 (0.88)
POMAP++	438	9	15.1	0.79	0.74	0.69
Holontology	318	9	16.8	0.80	0.83	0.87
DOME	13747	9	16.0	0.73	0.73	0.73
LogMap	14083	7	21.7	0.66	0.77 (0.80)	0.91 (1.00)
LogMapBio	2340	9	22.1	0.68	0.81	1.00
LogMapLt	500	6	22.0	0.61	0.72 (0.76)	0.87 (1.00)
Baseline	412	9	18.9	0.75	0.79	0.84
Property performance						
AML	88448	5	0.0	0.00	0.00	0.00
POMAP++	438	9	0.0	0.00	0.00	0.00
Holontology	318	9	0.0	0.00	0.00	0.00
DOME	13747	9	207.3	0.86	0.84	0.81
LogMap	14083	7	0.0	0.00	0.00	0.00
LogMapBio	2340	9	0.0	0.00	0.00	0.00
LogMapLt	500	6	0.0	0.00	0.00	0.00
Baseline	412	9	213.8	0.86	0.84	0.82
Instance performance						
AML	88448	5	82380.9	0.16	0.23 (0.26)	0.38 (0.63)
POMAP++	438	9	0.0	0.00	0.00	0.00
Holontology	318	9	0.0	0.00	0.00	0.00
DOME	13747	9	15688.7	0.61	0.61	0.61
LogMap	14083	7	97081.4	0.08	0.14 (0.15)	0.81 (0.93)
LogMapBio	2340	9	0.0	0.00	0.00	0.00
LogMapLt	500	6	82388.3	0.39	0.52 (0.56)	0.76 (0.96)
Baseline	412	9	17743.3	0.59	0.69	0.82
Overall performance						
AML	88448	5	102471.1	0.19	0.23 (0.28)	0.31 (0.52)
POMAP++	438	9	16.9	0.79	0.14	0.08
Holontology	318	9	18.8	0.80	0.17	0.10
DOME	13747	9	15912.0	0.68	0.68	0.67
LogMap	14083	7	97104.8	0.09	0.16 (0.16)	0.64 (0.74)
LogMapBio	2340	9	24.1	0.68	0.19	0.11
LogMapLt	500	6	88893.1	0.42	0.49 (0.54)	0.60 (0.77)
Baseline	412	9	17976.0	0.65	0.73	0.82

With respect to instance correspondences, AML, DOME, LogMap, LogMapLt were able to produce them (as was the baseline) whereas POMAP++, Holontology and LogMapBio were not, since they are not designed for instance matching. The baseline was unsurpassed by any system in this category in either F-measure or recall. One reason for this is that the baseline had the highest F-measure among systems able to match both classes and instances, and had a higher F-measure than DOME at matching properties, given that the alignment of instances is conditioned by the correct alignment of classes and properties. Furthermore, many of the matching systems return n:m correspondences and thus a lot of false positive correspondences, resulting in low precision.

We analyzed the errors for a specific task, namely `darkscape-oldschoolrunescape`. For this task, the baseline could not find the following correspondences: `Lumbridge_and_Draynor_Tasks = Lumbridge_&_Draynor_Diary` and `Cupric_sulphate = Cupric_sulfate`. The matcher AML does not find `Translated_notes = Translated_notes`, even if the label (wiki page name) is exactly the same. False positive correspondences for the LogMap matcher are `Ancient_Magicks = Carrallangar_Teleport` and `Ancient_Magicks = Kharyrll_Teleport`. For AML one example is `Customs_Officer = Gang_boss`.

Regarding runtime, AML was the slowest system, followed by DOME and LogMap. POMAP++ and Holontology were quite fast, but only return class correspondences.

4.12 Interactive matching

This year, the same four systems as last year participated in the Interactive matching track: ALIN, AML, LogMap, and XMap. Their results are shown in Table 17 and Figure 4 for both Anatomy and Conference datasets.

The table includes the following information (column names within parentheses):

- The performance of the system: Precision (Prec.), Recall (Rec.) and F-measure (F-m.) with respect to the fixed reference alignment, as well as Recall+ (Rec.+) for the Anatomy task. To facilitate the assessment of the impact of user interactions, we also provide the performance results from the original tracks, without interaction (line with Error NI).
- To ascertain the impact of the oracle errors, we provide the performance of the system with respect to the oracle (i.e., the reference alignment as modified by the errors introduced by the oracle: Precision oracle (Prec. oracle), Recall oracle (Rec. oracle) and F-measure oracle (F-m. oracle). For a perfect oracle these values match the actual performance of the system.
- Total requests (Tot Reqs.) represents the number of distinct user interactions with the tool, where each interaction can contain one to three conflicting correspondences, that could be analysed simultaneously by a user.
- Distinct correspondences (Dist. Mapps) counts the total number of correspondences for which the oracle gave feedback to the user (regardless of whether they were submitted simultaneously, or separately).
- Finally, the performance of the oracle itself with respect to the errors it introduced can be gauged through the positive precision (Pos. Prec.) and negative precision (Neg. Prec.), which measure respectively the fraction of positive and negative answers given by the oracle that are correct. For a perfect oracle these values are equal to 1 (or 0, if no questions were asked).

The figure shows the time intervals between the questions to the user/oracle for the different systems and error rates. Different runs are depicted with different colors.

Table 17. Interactive matching results for the Anatomy and Conference datasets.

Tool	Error	Prec.	Rec.	F-m.	Rec.+	Prec. oracle	Rec. oracle	F-m. oracle	Tot. Reqs.	Dist. Mapps	Pos. Prec.	Neg. Prec.
Anatomy Dataset												
ALIN	NI	0.998	0.611	0.758	0.0	–	–	–	–	–	–	–
	0.0	0.994	0.826	0.902	0.543	0.994	0.826	0.902	602	1448	1.0	1.0
	0.1	0.914	0.802	0.854	0.482	0.994	0.833	0.906	578	1373	0.731	0.965
	0.2	0.848	0.784	0.815	0.436	0.994	0.839	0.91	564	1343	0.561	0.931
	0.3	0.784	0.757	0.77	0.369	0.995	0.843	0.912	552	1307	0.419	0.875
AML	NI	0.95	0.936	0.943	0.832	–	–	–	–	–	–	–
	0.0	0.964	0.948	0.956	0.862	0.964	0.948	0.956	240	240	1.0	1.0
	0.1	0.952	0.946	0.948	0.857	0.965	0.95	0.957	268	268	0.719	0.97
	0.2	0.938	0.941	0.939	0.849	0.965	0.95	0.957	272	272	0.52	0.935
	0.3	0.92	0.938	0.929	0.843	0.966	0.951	0.958	299	299	0.379	0.905
LogMap	NI	0.918	0.846	0.88	0.593	–	–	–	–	–	–	–
	0.0	0.982	0.846	0.909	0.595	0.982	0.846	0.909	388	1164	1.0	1.0
	0.1	0.961	0.832	0.892	0.568	0.964	0.801	0.875	388	1164	0.742	0.966
	0.2	0.945	0.823	0.88	0.552	0.944	0.761	0.842	388	1164	0.567	0.927
	0.3	0.932	0.819	0.872	0.543	0.922	0.725	0.812	388	1164	0.434	0.878
XMap	NI	0.929	0.865	0.896	0.647	–	–	–	–	–	–	–
	0.0	0.929	0.867	0.897	0.653	0.929	0.867	0.897	35	35	1.0	1.0
	0.1	0.929	0.867	0.897	0.653	0.929	0.866	0.896	35	35	0.601	0.978
	0.2	0.929	0.867	0.897	0.653	0.929	0.865	0.896	35	35	0.4	0.965
	0.3	0.929	0.867	0.897	0.653	0.929	0.863	0.895	35	35	0.298	0.946
Conference Dataset												
ALIN	NI	0.88	0.456	0.601	–	–	–	–	–	–	–	–
	0.0	0.921	0.721	0.809	–	0.921	0.721	0.809	276	698	1.0	1.0
	0.1	0.725	0.686	0.705	–	0.934	0.753	0.834	264	674	0.538	0.987
	0.2	0.601	0.648	0.623	–	0.942	0.773	0.849	260	657	0.341	0.967
	0.3	0.495	0.624	0.552	–	0.951	0.796	0.866	259	645	0.226	0.95
AML	NI	0.841	0.659	0.739		–	–	–	–	–	–	–
	0.0	0.912	0.711	0.799	–	0.912	0.711	0.799	270	270	1.0	1.0
	0.1	0.838	0.698	0.762	–	0.923	0.733	0.817	277	277	0.691	0.971
	0.2	0.769	0.676	0.719	–	0.928	0.747	0.827	271	271	0.533	0.922
	0.3	0.715	0.663	0.688	–	0.931	0.758	0.836	270	270	0.459	0.885
LogMap	NI	0.818	0.59	0.686	–	–	–	–	–	–	–	–
	0.0	0.886	0.61	0.723	–	0.886	0.61	0.723	82	246	1.0	1.0
	0.1	0.85	0.596	0.7	–	0.858	0.576	0.69	82	246	0.71	0.978
	0.2	0.82	0.588	0.685	–	0.831	0.547	0.66	82	246	0.507	0.941
	0.3	0.793	0.583	0.672	–	0.808	0.518	0.631	82	246	0.366	0.907
XMap	NI	0.716	0.62	0.665	–	–	–	–	–	–	–	–
	0.0	0.719	0.62	0.666	–	0.719	0.62	0.666	16	16	0.0	1.0
	0.1	0.719	0.62	0.666	–	0.719	0.617	0.665	16	16	0.0	1.0
	0.2	0.718	0.62	0.666	–	0.72	0.613	0.662	16	16	0.2	1.0
	0.3	0.718	0.62	0.666	–	0.721	0.613	0.662	16	16	0.1	1.0

NI stands for non-interactive, and refers to the results obtained by the matching system in the original track.

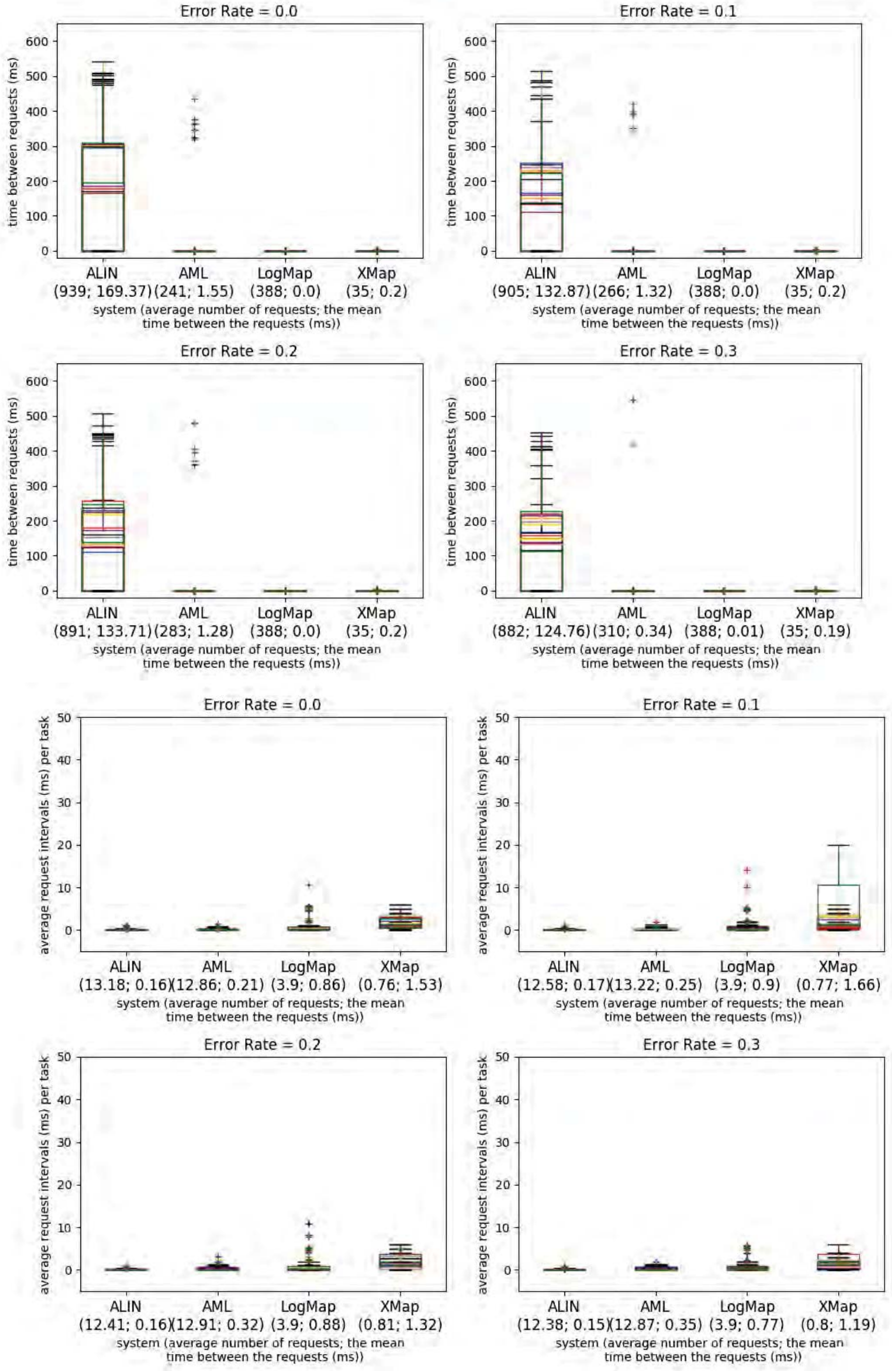


Fig. 4. Time intervals between requests to the user/oracle for the Anatomy (top 4 plots) and Conference (bottom 4 plots) datasets. Whiskers: Q1-1,5IQR, Q3+1,5IQR, IQR=Q3-Q1. The labels under the system names show the average number of requests and the mean time between the requests for the ten runs.

The matching systems that participated in this track employ different user-interaction strategies. While LogMap, XMap and AML make use of user interactions exclusively in the post-matching steps to filter their candidate correspondences, ALIN can also add new candidate correspondences to its initial set. LogMap and AML both request feedback on only selected correspondences candidates (based on their similarity patterns or their involvement in unsatisfiabilities) and AML presents one correspondence at a time to the user. XMap also presents one correspondence at a time and asks mainly about incorrect correspondences. ALIN and LogMap can both ask the oracle to analyze several conflicting correspondences simultaneously.

The performance of the systems usually improves when interacting with a perfect oracle in comparison with no interaction. The one exception is XMap, because it is barely interactive in the datasets. In general, XMap performs very few requests to the oracle compared to the other systems. Thus, it is also the system that improves the least with user interaction. On the other end of the spectrum, ALIN is the system that improves the most, because its high number of oracle requests and its non-interactive performance was the lowest of the interactive systems, and thus the easiest to improve.

Although system performance deteriorates when the error rate increases, there are still benefits from the user interaction—some of the systems' measures stay above their non-interactive values even for the larger error rates. Naturally, the more a system relies on the oracle, the more its performance tends to be affected by the oracle's errors.

The impact of the oracle's errors is linear for ALIN, AML and for XMap in most tasks, as the F-measure according to the oracle remains approximately constant across all error rates. It is supra-linear for LogMap in all datasets.

Another aspect that was assessed, was the response time of systems, i.e., the time between requests. Two models for system *response times* are frequently used in the literature [9]: Shneiderman and Seow take different approaches to categorize the response times taking a task-centered view and a user-centered view respectively. According to task complexity, Shneiderman defines response time in four categories: typing, mouse movement (50-150 ms), simple frequent tasks (1 s), common tasks (2-4 s) and complex tasks (8-12 s). While Seow's definition of response time is based on the user expectations towards the execution of a task: instantaneous (100-200 ms), immediate (0.5-1 s), continuous (2-5 s), captive (7-10 s). Ontology alignment is a cognitively demanding task and can fall into the third or fourth categories in both models. In this regard the response times (request intervals as we call them above) observed in all datasets fall into the tolerable and acceptable response times, and even into the first categories, in both models. The request intervals for AML, LogMap and XMAP stay at a few milliseconds for most datasets. ALIN's request intervals are higher, but still in the tenth of second range. It could be the case, however, that a user would not be able to take advantage of these low response times because the task complexity may result in higher user response time (i.e., the time the user needs to respond to the system after the system is ready).

4.13 Complex Matching

The only systems able to generate any kind of complex correspondence in any of the complex matching test cases were AMLC (in the Conference test suite) and CANARD (in the Taxon test case). No systems were capable of generating complex correspondences over either the Hydrography or the GeoLink test cases.

On the Conference test suite, only complex correspondences were being evaluated, since simple correspondences are already evaluated under the Conference track. In the case of the Hydrography and GeoLink test cases, all SEALS OAEI participants were evaluated in subtask 1 of both test cases, wherein they had to simply identify related entities. On the Taxon test case, all 14 systems which registered to the complex, conference and/or anatomy track were evaluated, but only 7 could output at least one alignment.

The results of the systems on the four test cases are summarized in Table 18.

Table 18. Results of the Complex Track. The precision, recall and F-measure are the average measures. QWR is the proportion of queries well rewritten.

Matcher	Conference			Hydrography (subtask 1)			GeoLink (subtask 1)			Taxon	
	Prec.	F-meas.	Rec.	Prec.	F-meas.	Rec.	Prec.	F-meas.	Rec.	Prec.	QWR
ABC	-	-	-	0.43	**0.18**	**0.12**	-	-	-	-	-
ALOD2Vec	-	-	-	0.5	0.09	0.05	0.78	**0.19**	**0.11**	-	-
AMLC	**0.54**	**0.42**	**0.34**	-	-	-	-	-	-	-	-
AML	-	-	-	-	-	-	-	-	-	0.00	0.00
CANARD	-	-	-	-	-	-	-	-	-	0.20	**0.13**
DOME	-	-	-	0.35	0.09	0.06	0.44	0.17	**0.11**	-	-
FMapX	-	-	-	**0.46**	0.11	0.07	-	-	-	-	-
Holontology	-	-	-	-	-	-	-	-	-	0.22	0.00
KEPLER	-	-	-	0.5	0.09	0.05	-	-	-	-	-
LogMap	-	-	-	0.44	0.08	0.05	0.85	0.18	0.1	**0.54**	0.07
LogMapBio	-	-	-	-	-	-	-	-	-	0.28	0.00
LogMapKG	-	-	-	-	-	-	0.85	0.18	0.1	-	-
LogMapLt	-	-	-	-	-	-	0.73	**0.19**	**0.11**	0.16	0.10
POMAP++	-	-	-	0.42	0.06	0.04	**0.9**	0.17	0.09	0.14	0.00
XMap	-	-	-	0.21	0.09	0.06	0.39	0.15	0.09	-	-

With respect to subtask 1 of the Hydrography and GeoLink test cases, the results show that a simple baseline approach that identifies target entity names within source entity comments performs better than most existing matchers. This is unsurprising, as matching systems are configured to find equivalent concepts rather than related ones. The takeaway from this year is that there is a lot of room for new approaches on this task.

In the Taxon test cases, only the output of LogMap, LogMapLt and CANARD could be used to rewrite source queries.

A more detailed discussion of the results of each task can be found in the OAEI page for this track. For a first edition of complex matching in an OAEI campaign, and given the inherent difficulty of the task, the results and participation are promising albeit still modest.

5 Conclusions & Lessons Learned

The OAEI 2018 counted this year several new tracks, some of which open new perspectives in the field, in particular with respect to the generation of more expressive alignments. We witnessed a slight decrease in the number of participants in comparison with previous years, but with a healthy mix of new and returning systems. However, like last year, the distribution of participants by tracks was uneven.

The **schema matching tracks** saw abundant participation, but, as has been the trend of the recent years, little substantial progress in terms of quality of the results or run time of top matching systems, judging from the long-standing tracks. On the one hand, this may be a sign of a performance plateau being reached by existing strategies and algorithms, which would suggest that new technology is needed to obtain significant improvements. On the other hand, it is also true that established matching systems tend to focus more on new tracks and datasets than on improving their performance in long-standing tracks, whereas new systems typically struggle to compete with established ones.

The number of matching systems capable of handling very large ontologies has increased slightly over the last years, but is still relatively modest, judging from the *Large Biomedical Ontologies* track. We will aim at facilitating participation in future editions of this track by providing techniques to divide the matching tasks in manageable subtasks (e.g., [27]).

There has also been progress, but likewise room for improvement, on the ability of matching systems to match properties, judging from the *Conference* track. To assist system developers in tackling this aspect, we plan to provide a more detailed evaluation in the future, including an analysis of the false positives per matching system.

Less encouraging is the low number of systems concerned with the logical coherence of the alignments they produce, an aspect which is critical for several semantic web applications. Perhaps a more direct approach is needed to promote this topic, such as providing a more in-depth analysis of the causes of incoherence in the evaluation or even organizing a future track focusing on logical coherence alone.

The consensus-based evaluation in the *Disease and Phenotype* track offers limited insights into performance, as several matching systems produce a number of unique correspondences which may or may not be correct. In the absence of a true reference alignment, future evaluation should seek to determine whether the unique correspondences contain indicators of correctness, such as semantic similarity, or appear to be noise.

The **instance matching tracks** and the new **instance and schema matching track** counted few participants, as has been the trend in recent years. Part of the reason for this is that several of these tracks ran on the HOBBIT platform, and the transition from SEALS to HOBBIT has not been as easy as we might desire. Thus, participation should increase next year as systems become more familiar with the HOBBIT platform and have more time to do the migration. Furthermore, from an infrastructure point of view, the HOBBIT SDK will make the developing and debugging phase easier, and the Maven-based framework will facilitate submission. However, another factor behind the reduced participation in the instance matching tracks lies with their specialization. New schema matching tracks such as *Biodiversity and Ecology* typically demand very little from systems that are already able to tackle long-standing tracks such as *Anatomy*, whereas instance matching tracks such as *IIMB*, *Link Discovery* and last year's *Process Model Matching*, are so different from one another that each requires dedicated development time to tackle. Thus, in future OAEI editions we should consider publishing new instance matching (and other more specialized) datasets with more time in advance, to give system developers adequate time to tackle them. Equally critical will be to ensure stability by maintaining instance matching tracks and datasets over multiple OAEI editions, so that participants can build upon the development of previous years.

Automatic instance-matching benchmark generation algorithms have been gaining popularity, as evidenced by the fact that they are used in all three instance matching tracks of this OAEI edition. One aspect that has not been addressed in such algorithms is that, if the transformation is too extreme, the correspondence may be unrealistic and impossible to detect even by humans. As such, we argue that *human-in-the-loop* techniques can be exploited to do a preventive quality-checking of generated correspondences, and refine the set of correspondences included in the final reference alignment. We will explore such an approach in future editions of the *IIMB* track.

The **interactive matching track** also witnessed a small number of participants, which have been the same 4 systems over the last three campaigns. This is puzzling considering that this track is based on the *Anatomy* and *Conference* test cases, and those tracks had 14 participants. The process of programmatically querying the Oracle class used to simulate user interactions is simple enough that it should not be a deterrent for participation, but perhaps we should look at facilitating the process further in future OAEI editions by providing implementation examples.

Finally, the **complex matching track** opens new perspectives in the field of ontology matching, as this is a topic largely unexplored but of growing importance, since integrating linked datasets often encompasses making complex correspondences. Tackling complex matching automatically is extremely challenging, likely requiring profound adaptations from matching systems, so the fact that there were two participants able to generate complex correspondences in this track should be seen as a positive sign of progress to the state of the art in ontology matching. While this year the track involved different evaluation settings, we will work towards enabling the automatic evaluation of complex alignments in future editions.

Like in previous OAEI editions, most participants provided a description of their systems and their experience in the evaluation, in the form of OAEI system papers. These papers, like the present one, have not been peer reviewed. However, they are full contributions to this evaluation exercise, reflecting the effort and insight of matching systems developers, and providing details about those systems and the algorithms they implement.

The Ontology Alignment Evaluation Initiative will strive to remain a reference to the ontology matching community by improving both the test cases and the testing methodology to better reflect actual needs, as well as to promote progress in this field [41]. 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 to have 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 papers that follow.

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

We are also grateful to Martin Ringwald and Terry Hayamizu for providing the reference alignment for the anatomy ontologies and thank Elena Beisswanger for her thorough support on improving the quality of the dataset.

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

We thank Andrea Turbati and the AGROVOC team for their very appreciated help with the preparation of the AGROVOC subset ontology. We are also grateful to Catherine Roussey and Nathalie Hernandez for their help on the Taxon alignment.

We also thank for their support the past members of the Ontology Alignment Evaluation Initiative steering committee: Jérôme Euzenat (INRIA, FR), Yannis Kalfoglou (Ricoh laboratories, UK), Miklos Nagy (The Open University,UK), Natasha Noy (Google Inc., 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).

Cássia Trojahn dos Santos has been partially supported by the French CIMI Labex project IBLiD (Integration of Big and Linked Data for On-Line Analytics).

Daniel Faria was supported by the ELIXIR-EXCELERATE project (INFRADEV-3-2015).

Ernesto Jimenez-Ruiz has been partially supported by the BIGMED project (IKT 259055), the SIRIUS Centre for Scalable Data Access (Research Council of Norway, project no.: 237889), and the AIDA project (UK Government's Defence & Security Programme in support of the Alan Turing Institute.

Catia Pesquita was supported by the FCT through the LASIGE Strategic Project (UID/CEC/00408/2013) and the research grant PTDC/EEI-ESS/4633/2014.

Irini Fundulaki and Tzanina Saveta were supported by the European Union's Horizon 2020 research and innovation programme under grant agreement No 688227 (Hobbit).

Jana Vataščinová and Ondřej Zamazal have been supported by the CSF grant no. 18-23964S.

Patrick Lambrix and Huanyu Li have been supported by the Swedish e-Science Research Centre and the Swedish National Graduate School in Computer Science (CUGS).

References

1. Manel Achichi, Michelle Cheatham, Zlatan Dragisic, Jérôme Euzenat, Daniel Faria, Alfio Ferrara, Giorgos Flouris, Irini Fundulaki, Ian Harrow, Valentina Ivanova, Ernesto Jiménez-

Ruiz, Kristian Kolthoff, Elena Kuss, Patrick Lambrix, Henrik Leopold, Huanyu Li, Christian Meilicke, Majid Mohammadi, Stefano Montanelli, Catia Pesquita, Tzanina Saveta, Pavel Shvaiko, Andrea Splendiani, Heiner Stuckenschmidt, Élodie Thiéblin, Konstantin Todorov, Cássia Trojahn, and Ondrej Zamazal. Results of the ontology alignment evaluation initiative 2017. In *Proceedings of the 12th International Workshop on Ontology Matching, Vienna, Austria, October 21*, pages 61–113, 2017.

2. Manel Achichi, Michelle Cheatham, Zlatan Dragisic, Jerome Euzenat, Daniel Faria, Alfio Ferrara, Giorgos Flouris, Irini Fundulaki, Ian Harrow, Valentina Ivanova, Ernesto Jiménez-Ruiz, Elena Kuss, Patrick Lambrix, Henrik Leopold, Huanyu Li, Christian Meilicke, Stefano Montanelli, Catia Pesquita, Tzanina Saveta, Pavel Shvaiko, Andrea Splendiani, Heiner Stuckenschmidt, Konstantin Todorov, Cássia Trojahn, and Ondrej Zamazal. Results of the ontology alignment evaluation initiative 2016. In *Proceedings of the 11th International Ontology matching workshop, Kobe (JP), October 18th*, pages 73–129, 2016.
3. 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 *Proceedings of the 7th International Ontology matching workshop, Boston (MA, US)*, pages 73–115, 2012.
4. Benhamin Ashpole, Marc Ehrig, Jérôme Euzenat, and Heiner Stuckenschmidt, editors. *Proc. K-Cap Workshop on Integrating Ontologies*, Banff (Canada), 2005.
5. Olivier Bodenreider. The unified medical language system (UMLS): integrating biomedical terminology. *Nucleic Acids Research*, 32:267–270, 2004.
6. 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 *Proceedings of the 3rd Ontology matching workshop, Karlsruhe (DE)*, pages 73–120, 2008.
7. Michelle Cheatham, Zlatan Dragisic, Jérôme Euzenat, Daniel Faria, Alfio Ferrara, Giorgos Flouris, Irini Fundulaki, Roger Granada, Valentina Ivanova, Ernesto Jiménez-Ruiz, Patrick Lambrix, Stefano Montanelli, Catia Pesquita, Tzanina Saveta, Pavel Shvaiko, Alessandro Solimando, Cássia Trojahn, and Ondřej Zamazal. Results of the ontology alignment evaluation initiative 2015. In *Proceedings of the 10th International Ontology matching workshop, Bethlehem (PA, US)*, pages 60–115, 2015.
8. 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, *Proceedings of the 8th International Ontology matching workshop, Sydney (NSW, AU)*, pages 61–100, 2013.
9. Jim Dabrowski and Ethan V. Munson. 40 years of searching for the best computer system response time. *Interacting with Computers*, 23(5):555–564, 2011.
10. Thaleia Dimitra Doudali, Ioannis Konstantinou, and Nectarios Koziris Doudali. Spaten: a Spatio-Temporal and Textual Big Data Generator. In *IEEE Big Data*, pages 3416–3421, 2017.
11. Zlatan Dragisic, Kai Eckert, Jérôme Euzenat, Daniel Faria, Alfio Ferrara, Roger Granada, Valentina Ivanova, Ernesto Jiménez-Ruiz, Andreas Oskar Kempf, Patrick Lambrix, Stefano Montanelli, Heiko Paulheim, Dominique Ritze, Pavel Shvaiko, Alessandro Solimando, Cássia Trojahn dos Santos, Ondrej Zamazal, and Bernardo Cuenca Grau. Results of the on-

tology alignment evaluation initiative 2014. In *Proceedings of the 9th International Ontology matching workshop, Riva del Garda (IT)*, pages 61–104, 2014.

12. Zlatan Dragisic, Valentina Ivanova, Patrick Lambrix, Daniel Faria, Ernesto Jiménez-Ruiz, and Catia Pesquita. User validation in ontology alignment. In *Proceedings of the 15th International Semantic Web Conference, Kobe, Japan, October 17-21*, pages 200–217, 2016.
13. Zlatan Dragisic, Valentina Ivanova, Huanyu Li, and Patrick Lambrix. Experiences from the anatomy track in the ontology alignment evaluation initiative. *Journal of Biomedical Semantics*, 8:56:1–56:28, 2017.
14. 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 *Proceedings of the 4th International Ontology matching workshop, Chantilly (VA, US)*, pages 73–126, 2009.
15. 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 *Proceedings of the 5th International Ontology matching workshop, Shanghai (CN)*, pages 85–117, 2010.
16. 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 *Proceedings of the 6th International Ontology matching workshop, Bonn (DE)*, pages 85–110, 2011.
17. 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 *Proceedings 2nd International Ontology matching workshop, Busan (KR)*, pages 96–132, 2007.
18. 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.
19. 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 *Proceedings of the 1st International Ontology matching workshop, Athens (GA, US)*, pages 73–95, 2006.
20. Jérôme Euzenat and Pavel Shvaiko. *Ontology matching*. Springer-Verlag, Heidelberg (DE), 2nd edition, 2013.
21. Daniel Faria, Ernesto Jiménez-Ruiz, Catia Pesquita, Emanuel Santos, and Francisco M. Couto. Towards Annotating Potential Incoherences in BioPortal Mappings. In *Proceedings of the 13th International Semantic Web Conference*, volume 8797, pages 17–32. Springer, 2014.
22. Alfio Ferrara, Stefano Montanelli, Jan Noessner, and Heiner Stuckenschmidt. Benchmarking matching applications on the semantic web. In *Proceedings of the 8th Extended Semantic Web Conference*, Heraklion, Greece, 2011.
23. Ian Harrow, Ernesto Jiménez-Ruiz, Andrea Splendiani, Martin Romacker, Peter Woollard, Scott Markel, Yasmin Alam-Faruque, Martin Koch, James Malone, and Arild Waaler. Matching Disease and Phenotype Ontologies in the Ontology Alignment Evaluation Initiative. *Journal of Biomedical Semantics*, 2017.
24. Sven Hertling and Heiko Paulheim. Dbkwik: A consolidated knowledge graph from thousands of wikis. In *Proceedings of the International Conference on Big Knowledge*, 2018.

25. Alexandra Hofmann, Samresh Perchani, Jan Portisch, Sven Hertling, and Heiko Paulheim. Dbkwik: Towards knowledge graph creation from thousands of wikis. In *Proceedings of the International Semantic Web Conference (Posters and Demos), Vienna, Austria*, pages 21–25, 2017.
26. Valentina Ivanova, Patrick Lambrix, and Johan Åberg. Requirements for and evaluation of user support for large-scale ontology alignment. In *Proceedings of the European Semantic Web Conference*, pages 3–20, 2015. doi: 10.1007/978-3-319-18818-8_1.
27. Ernesto Jiménez-Ruiz, Asan Agibetov, Matthias Samwald, and Valerie Cross. Breaking-down the Ontology Alignment Task with a Lexical Index and Neural Embeddings. *CoRR*, abs/1805.12402, 2018.
28. Ernesto Jiménez-Ruiz and Bernardo Cuenca Grau. LogMap: Logic-based and scalable ontology matching. In *Proceedings of the 10th International Semantic Web Conference, Bonn (DE)*, pages 273–288, 2011.
29. 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.
30. Ernesto Jiménez-Ruiz, Christian Meilicke, Bernardo Cuenca Grau, and Ian Horrocks. Evaluating mapping repair systems with large biomedical ontologies. In *Proceedings of the 26th Description Logics Workshop*, 2013.
31. Ernesto Jiménez-Ruiz, Tzanina Saveta, Ondvrej Zamazal, Sven Hertling, Michael Röder, Irini Fundulaki, Axel-Cyrille Ngonga Ngomo, Mohamed Ahmed Sherif, Amina Annane, Zohra Bellahsene, Sadok Ben Yahia, Gayo Diallo, Daniel Faria, Marouen Kachroudi, Abderrahmane Khiat, Patrick Lambrix, Huanyu Li, Maximilian Mackeprang, Majid Mohammadi, Maciej Rybinski, Booma Sowkarthiga Balasubramani, and Cassia Trojahn. Introducing the HOBBIT platform into the Ontology Alignment Evaluation Campaign. In *Proceedings of the 13th International Workshop on Ontology Matching*, 2018.
32. Naouel Karam, Claudia Müller-Birn, Maren Gleisberg, David Fichtmüller, Robert Tolksdorf, and Anton Güntsch. A terminology service supporting semantic annotation, integration, discovery and analysis of interdisciplinary research data. *Datenbank-Spektrum*, 16(3):195–205, 2016.
33. Yevgeny Kazakov, Markus Krötzsch, and Frantisek Simancik. Concurrent classification of EL ontologies. In *Proceedings of the 10th International Semantic Web Conference, Bonn (DE)*, pages 305–320, 2011.
34. Christian Meilicke. *Alignment Incoherence in Ontology Matching*. PhD thesis, University Mannheim, 2011.
35. 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.
36. Boris Motik, Rob Shearer, and Ian Horrocks. Hypertableau reasoning for description logics. *Journal of Artificial Intelligence Research*, 36:165–228, 2009.
37. Heiko Paulheim, Sven Hertling, and Dominique Ritze. Towards evaluating interactive ontology matching tools. In *Proceedings of the 10th Extended Semantic Web Conference, Montpellier (FR)*, pages 31–45, 2013.
38. Manuel Salvadores, Paul R. Alexander, Mark A. Musen, and Natalya Fridman Noy. BioPortal as a dataset of linked biomedical ontologies and terminologies in RDF. *Semantic Web*, 4(3):277–284, 2013.
39. Emanuel Santos, Daniel Faria, Catia Pesquita, and Francisco M Couto. Ontology alignment repair through modularization and confidence-based heuristics. *PLoS ONE*, 10(12):e0144807, 2015.

40. Tzanina Saveta, Evangelia Daskalaki, Giorgos Flouris, Irini Fundulaki, Melanie Herschel, and Axel-Cyrille Ngonga Ngomo. Pushing the limits of instance matching systems: A semantics-aware benchmark for linked data. In *Proceedings of the 24th International Conference on World Wide Web*, pages 105–106, New York, NY, USA, 2015. ACM.
41. 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.
42. Alessandro Solimando, Ernesto Jiménez-Ruiz, and Giovanna Guerrini. Detecting and correcting conservativity principle violations in ontology-to-ontology mappings. In *Proceedings of the International Semantic Web Conference*, pages 1–16. Springer, 2014.
43. Alessandro Solimando, Ernesto Jimenez-Ruiz, and Giovanna Guerrini. Minimizing conservativity violations in ontology alignments: Algorithms and evaluation. *Knowledge and Information Systems*, 2016.
44. Christian Strobl. *Encyclopedia of GIS*, chapter Dimensionally Extended Nine-Intersection Model (DE-9IM), pages 240–245. Springer, 2008.
45. York Sure, Oscar Corcho, Jérôme Euzenat, and Todd Hughes, editors. *Proceedings of the Workshop on Evaluation of Ontology-based Tools (EON), Hiroshima (JP)*, 2004.
46. Elodie Thiéblin, Michelle Cheatham, Cassia Trojahn, Ondrej Zamazal, and Lu Zhou. The First Version of the OAEI Complex Alignment Benchmark. In *Proceedings of the International Semantic Web Conference (Posters and Demos)*, 2018.
47. Ondřej Zamazal and Vojtěch Svátek. The ten-year ontofarm and its fertilization within the onto-sphere. *Web Semantics: Science, Services and Agents on the World Wide Web*, 43:46–53, 2017.
48. Lu Zhou, Michelle Cheatham, Adila Krisnadhi, and Pascal Hitzler. A complex alignment benchmark: Geolink dataset. In *Proceedings of the 17th International Semantic Web Conference, Monterey (CA, USA), October 8-12*, pages 273–288, 2018.

Jena, Dayton, Lisboa, Milano, Heraklion,
Mannheim, Oslo, Berlin, Linköping, Porto Alegre,
Trento, Toulouse, Prague, London
November 2018

ALIN Results for OAEI 2018

Jomar da Silva[1], Kate Revoredo[1], and Fernanda Araujo Baião[1]

Graduated Program in Informatics, Department of Applied Informatics
Federal University of the State of Rio de Janeiro (UNIRIO), Brazil
{jomar.silva,katerevoredo,fernanda.baiao}@uniriotec.br

Abstract. ALIN is an ontology matching system specialized in the interactive ontology matching, and its main characteristic is the use of expert feedback to improve the set of mapping suggestions, using semantic and structural techniques to make this improvement. ALIN has obtained the alignment with the highest quality in the interactive tracking for Conference data set. This paper describes its configuration for the OAEI 2018 competition and discusses its results.

Keywords: ontology matching, Wordnet, interactive ontology matching, ontology alignment, interactive ontology alignment

1 Presentation of the system

Due to the advances in information and communication technologies, a large amount of data repositories became available. Those repositories, however, are highly semantically heterogeneous, which hinders their integration. Ontology matching has been successfully applied to solve this problem, by discovering mappings between two distinct ontologies which, in turn, conceptually define the data stored in each repository. Among the various ontology matching approaches that exist in the literature, interactive ontology matching includes the participation of domain experts to improve the quality of the final alignment [1]. ALIN is an interactive ontology matching system and has participated in the OAEI 2016 and OAEI 2017 evaluations.

1.1 State, purpose, general statement

ALIN has the following steps to perform the interactive ontology matching process: First, ALIN generates an initial set of mappings. This set is called the set of mapping suggestions, that are the mappings to receive expert feedback. After, the interactive phase begins, where, at each interaction, the expert gives his feedback for some mapping suggestions. After each expert feedback, ALIN modifies the set of mapping suggestions according to the expert feedback. The modification of the set of mapping suggestions is by the use of the structural analysis of ontologies and the use of alignment anti-patterns. The interactions continue until there are no more mapping suggestions left.

Algorithm 1 ALIN algorithm

```
Input: Two ontologies to be aligned, similarity metrics
Output: Alignment between the two ontologies
 1: Loading of ontologies
 2: Generation of the initial set of mapping suggestions
 3: Move of mappings by automatic classification from the set of mappings suggestions to the alignment
 4: Move of mappings by the low value of semantic similarity from the set of mapping suggestions to a backup set
 5: while Set of mapping suggestions is not empty do
 6:     Choose mapping from the the set of mapping suggestions to submit to the expert
 7:     Receive expert feedback to chosen mapping and remove it from the set of mapping suggestions
 8:     if Mapping is accepted then
 9:         Remove mappings in an alignment anti-pattern with accepted mapping from the set of mapping suggestions
10:         Insert some data property and object property mappings related to the accepted mapping into set of mapping suggestions
11:         Move some mappings related to the accepted mapping from the backup set to the set of mapping suggestions
12:     end if
13: end while
```

1.2 Specific techniques used

The steps of ALIN algorithm (Algorithm 1) are the following:

- Line 1. ALIN loads the ontology classes, object properties, and data properties through the Alignment API [2]. For each entity, some data are stored such as name and label. ALIN saves the class superclasses and disjunctions. ALIN also saves information about the object properties, like their hypernyms and their associated classes, and information about the data properties, like their associated class. ALIN does not use instances. The ALIN can only work with ontologies whose entity names are in English.
- Line 2. For each similarity metric, ALIN finds a set of mappings using a simple matching algorithm. The simple matching algorithm treats the matching problem as a stable marriage problem with size list limited to 1 [3], i.e., the algorithm only selects one mapping if similarity value between the two entities of the mapping is the highest considering all the mappings with at least one of these entities. ALIN uses six metrics and runs six times, once for each one, giving rise, each execution, to a set of mappings. The union of the sets gives origin to the initial set of mapping suggestions. ALIN uses the linguist metrics Jaccard, Jaro-Winkler, n-Gram, Resnick, Jiang-Conrath, and Lin. Simmetrics API [4] provides the metrics Jaccard, Jaro-Winkler, and n-Gram

and HESML API [5] the metrics Resnick, Jiang-Conrath, and Lin. HESML API uses Wordnet. Because ALIN needs the canonical form of the entity names to use the Wordnet, ALIN uses Stanford CoreNLP API [6]. ALIN uses the most frequent synsets of words to calculate semantic similarities.

- Line 3. The value of the similarity metrics (Resnick, Jiang-Conrath, Lin, Jaccard, Jaro-Winkler, and n-Gram) varies from 0 to 1 (1 is the maximum value). When one mapping in the set of mapping suggestions has all the six metrics with the maximum value, ALIN moves the mapping from the set of mapping suggestions to the final alignment.
- Line 4. ALIN moves the mappings whose entities has one of its linguistic metrics less than a given threshold from the set of mapping suggestions to a backup set. These mappings can return later, by structural analysis, to the set of mapping suggestions. [7] shows this technique, but with a little difference, it didn't use a threshold. It moves the class mappings that are not in the same Wordnet synset.
- Lines from 5 to 13. At this point, the interactions with the expert begin. ALIN sorts the mappings in the set of mapping suggestions by the sum of similarity metric values, greater sum first. ALIN submits the mappings to the expert. The set of mapping suggestions has, at first, only class mappings. After each expert feedback, if the expert accepts the mapping, ALIN moves it from the set of mapping suggestions to the alignment, else ALIN removes it from the set of mapping suggestions. ALIN can remove mappings (besides the mappings that received feedback) from the set of mapping suggestions and can include other mappings into it, depending on the expert feedback.

At each interaction with the expert:

- ALIN removes from the set of mapping suggestions all the mappings that are in alignment anti-pattern [8][9] with the accepted mapping;
- ALIN inserts into the set of mapping suggestions, data property (like [10]) and object property mappings related to the accepted class mappings.
- ALIN moves from the backup set to the set of mapping suggestions all mappings whose both entities are subclasses of the classes of an accepted mapping. [7] shows a similar technique.

The interaction phase continues until the set of mapping suggestions is empty.

1.3 Link to the system and parameters file

ALIN is available through Google drive (https://drive.google.com/file/d/1v6cxQvAuWVqIBzWQUEIzDuZogoW35fdq/view?usp=sharing) as a package for running through the SEALS client.

2 Results

Interactive ontology matching is the focus of the ALIN system. The quality of the alignment generated by ALIN is dependent on the correct expert feedback. ALIN has two phases: the non-interactive and the interactive phases. The non-interactive phase goal is to achieve high precision without worrying about the recall. In the interactive phase, ALIN modifies the set of mapping suggestions, including and removing mappings related to the accepted mapping. If the expert makes a mistake, ALIN is more prone to deteriorate the set of mapping suggestions, thereby decreasing the F-measure.

The system performs better when the number of data and object properties documented in the ontologies is proportionately large. In the interactive phase, the system includes into the set of mapping suggestions mappings related with accepted class mappings, thus allowing increase the recall. When the number of properties in the ontologies is small, the system still generates an alignment with good precision, but its recall tends to be not so good.

2.1 Comments on the participation of the ALIN in non-interactive tracks

As expected the participation of ALIN in non-interactive matching tracks showed the following results: high precision and not so high recall when compared to the other tools, as can be seen in Anatomy track[1] (Table 1). The conference track results can be seen on the OAEI 2018[2] page.

2.2 Comments on the participation of the ALIN in interactive tracks

Interactive Anatomy Track In this track, the program ALIN showed the highest precision among the four evaluated tools when the error rate is zero (Table 2). When the error rate increases, both the precision as the recall falls, so falling the F-measure (Table 3). Dependence on expert feedback to ensure precision and to increase recall explains this decline in quality when the expert makes mistakes.

As ontologies of the Anatomy Track contains almost no properties, ALIN cannot utilize some interactive techniques like the selection of property mappings related to accepted class mappings. Not using these techniques has limited the increase in recall, which influenced the F-measure.

---

[1] Results for OAEI 2018 - Anatomy track. Available at http://oaei.ontologymatching.org/2018/results/anatomy/ Last accessed on Oct, 02, 2018.

[2] Results of Evaluation for the Conference track within OAEI 2018 . Available at http://oaei.ontologymatching.org/2018/results/conference/index.html Last accessed on Oct, 24, 2018.

Table 1. Participation of ALIN in Anatomy non-interactive track

Tool	Precision	Recall	F-Measure
AML	0.95	0.936	0.943
LogMapBio	0.888	0.908	0.898
POMAP++	0.919	0.877	0.897
XMap	0.929	0.865	0.896
LogMap	0.918	0.846	0.88
SANOM	0.888	0.844	0.865
FCAMapX	0.941	0.791	0.859
KEPLER	0.958	0.741	0.836
Lily	0.872	0.795	0.832
LogMapLite	0.962	0.728	0.828
ALOD2Vec	0.996	0.648	0.785
StringEquiv	0.997	0.622	0.766
DOME	0.997	0.615	0.761
ALIN	0.998	0.611	0.758
Holontology	0.976	0.294	0.451

Table 2. Participation of ALIN in Anatomy interactive track - Error rate 0.0

Tool	Precision	Recall	F-measure	Total Requests
ALIN	0.994	0.826	0.902	602
AML	0.964	0.948	0.956	240
LogMap	0.982	0.846	0.909	388
XMap	0.929	0.867	0.897	35

Table 3. Participation of ALIN in Anatomy interactive track - Error rate 0.1

Tool	Precision	Recall	F-measure	Total Requests
ALIN	0.914	0.802	0.854	578
AML	0.952	0.946	0.948	268
LogMap	0.961	0.832	0.892	388
XMap	0.929	0.867	0.897	35

Interactive Conference Track In this track, ALIN stood out, showing the greatest F-measure among the four tools when the error rate is zero (Table 4), as with a loss of F-measure when the error rate increases (Table 5).

Table 4. Participation of ALIN in Conference interactive track - Error rate 0.0

Tool	Precision	Recall	F-measure	Total Requests
ALIN	0.921	0.721	0.809	276
AML	0.912	0.711	0.799	270
LogMap	0.886	0.61	0.723	82
XMap	0.719	0.62	0.666	16

Table 5. Participation of ALIN in Conference interactive track - Error rate 0.1

Tool	Precision	Recall	F-measure	Total Requests
ALIN	0.725	0.686	0.705	264
AML	0.838	0.698	0.762	277
LogMap	0.85	0.596	0.7	82
XMap	0.719	0.62	0.666	16

Other results, including results with different error rates, can be seen on the OAEI 2018[3] page.

2.3 Comparison of the participation to ALIN in OAEI 2018 with his participation in OAEI 2017

- One modification made in ALIN was the withdrawal of additional criteria for the automatic classification of mappings. At the beginning of its execution, ALIN automatically selects mappings with the entities with the same name to put into the alignment. In the OAEI 2017, ALIN used additional criteria for that, that is, if a mapping had the two entities with the same name, but had met one of those criteria, ALIN didn't put it into the alignment. In the conference data set, the use of these criteria increased the precision of the alignment, and thus its quality, but also the number of interactions. In the Anatomy data set, the use of these criteria increased only the number of interactions. For OAEI 2018, ALIN focused on reducing its number of interactions. So, ALIN doesn't use the additional criteria for the automatic classification of mappings anymore. This modification reduced the number of interactions (Total Requests) in both the anatomy track (Table 6) and the conference track (Table 7), without decreasing the quality (F-measure) on the anatomy track.
- Another modification was the selection of new mappings to the set of mapping suggestions. For OAEI, one interactive matching system can place up to three related mappings in an interaction. To take advantage of this rule,

---

[3] Results for OAEI 2018 - Interactive Track . Available at http://oaei.ontologymatching.org/2018/results/interactive/ Last accessed on Oct, 2, 2018.

in 2018, ALIN selects new mappings, with at least one entity equal to other already selected, to put into the set of mapping suggestions. This selection increases the likelihood of raising the recall. This modification increased the recall on the anatomy track (Table 6) but not increased enough on the conference track (Table 7) to compensate for the first modification.
- ALIN has stopped using the WS4J API[4]. ALIN had already stopped using WS4J to calculate similarity in OAEI 2017, starting to use HESML. ALIN was only using WS4J to find the most common synset to an entity name, but now ALIN is directly accessing the Wordnet files.

Table 6. Participation of ALIN in Anatomy interactive track - OAEI 2016[11]/2017[12]/2018- Error rate 0.0

Year	Precision	Recall	F-measure	Total Requests
2016	0.993	0.749	0.854	803
2017	0.993	0.794	0.882	939
2018	0.994	0.826	0.902	602

Table 7. Participation of ALIN in Conference interactive track - OAEI 2016[11]/2017[12]/2018- Error rate 0.0

Year	Precision	Recall	F-measure	Total Requests
2016	0.957	0.735	0.831	326
2017	0.957	0.731	0.829	329
2018	0.921	0.721	0.809	276

3 General Comments

Evaluating the results it can be seen that the system can be improved towards:

- handling user error rate;
- generating a higher quality (especially w.r.t. recall) initial alignment in its non-interactive phase;
- reducing the number of interactions with the expert.

---

[4] 'WS4J'. Available at https://github.com/Sciss/ws4j Last accessed on Jan, 16, 2018.

3.1 Conclusions

The ALIN system stands out in the interactive ontology matching process when ontologies have some characteristics, such as many documented properties, and when the expert does not make mistakes.

The second author was partially funding by project PQ-UNIRIO N01/2017 (" Aprendendo, adaptando e alinhando ontologias:metodologias e algoritmos.") and CAPES/PROAP.

References

1. Paulheim, H., Hertling, S., Ritze, D.: Towards Evaluating Interactive Ontology Matching Tools. Lecture Notes in Computer Science **7882** (2013) 31–45
2. David, J., Euzenat, J., Scharffe, F., Trojahn dos Santos, C.: The alignment api 4.0. Semant. web **2**(1) (January 2011) 3–10
3. Irving, R.W., Manlove, D.F., O'Malley, G.: Stable marriage with ties and bounded length preference lists. Journal of Discrete Algorithms **7**(2) (2009) 213 – 219 Selected papers from the 2nd Algorithms and Complexity in Durham Workshop ACiD 2006.
4. Surhone, L.M., Timpledon, M.T., Marseken, S.F.: SimMetrics. VDM Publishing (2010)
5. Lastra-Daz, J.J., Garca-Serrano, A., Batet, M., Fernndez, M., Chirigati, F.: Hesml. Inf. Syst. **66**(C) (June 2017) 97–118
6. Manning, C.D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S.J., McClosky, D.: The Stanford CoreNLP natural language processing toolkit. In: Association for Computational Linguistics (ACL) System Demonstrations. (2014) 55–60
7. Silva, J., Baião, F., Revoredo, K., Euzenat, J.: Semantic interactive ontology matching: Synergistic combination of techniques to improve the set of candidate correspondences. In: OM-2017: Proceedings of the Twelfth International Workshop on Ontology Matching. Volume 2032. (2017) 13–24
8. Guedes, A., Baião, F., Shivaprabhu, Revoredo, R.: On the Identification and Representation of Ontology Correspondence Antipatterns. In: Proc. 5th Int. Conf. Ontol. Semant. Web Patterns (WOP14), CEUR Work. Proc. (2014)
9. Guedes, A., Baião, F., Revoredo, K.: Digging Ontology Correspondence Antipatterns. In: Proceeding WOP14 Proc. 5th Int. Conf. Ontol. Semant. Web Patterns. Volume 1032. (2014) 38–48
10. Silva, J., Revoredo, K., Baião, F.A., Euzenat, J.: Interactive Ontology Matching: Using Expert Feedback to Select Attribute Mappings. (2018)
11. Silva, J., Baião, F., Revoredo, K.: Alin results for oaei 2016. In: OM-2016: Proceedings of the Eleventh International Workshop on Ontology Matching. OM'16 (2016) 130–137
12. Silva, J., Baião, F., Revoredo, K.: Alin results for oaei 2017. In: OM-2017: Proceedings of the Twelfth International Workshop on Ontology Matching. OM'17 (2017) 114–121

Results of AML participation in OAEI 2018

Daniel Faria[1], Catia Pesquita[2], Booma Sowkarthiga Balasubramani[3],
Teemu Tervo[2], David Carriço[2], Rodrigo Garrilha[2],
Francisco M. Couto[2], and Isabel F. Cruz[3]

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

Abstract. AgreementMakerLight (AML) is a system for automated ontology matching that is characterized by its efficiency, extensibility, and ability to incorporate external knowledge. In OAEI 2018, AML leveraged these features to expand its capabilities to tackle the new tracks. Particular effort was put into extending AML to produce complex mappings, and into improving instance matching approaches. AML was the only system to participate in all OAEI tracks this year, and was the top performing system, or among the top performing systems, in most tracks.

1 Presentation of the System

1.1 State, Purpose, General Statement

AgreementMakerLight (AML) is an ontology matching system based on the design principles of AgreementMaker [1, 2] with an added focus on efficiency, to be able to tackle large-scale ontology matching problems [7]. Its initial focus was the biomedical domain, but it has been continually expanded to address a broad range of ontology and instance matching problems, and it is now a general purpose ontology matching system. AML relies primarily on lexical matching algorithms [8], but also includes structural algorithms for both matching and filtering, as well as its own logical repair algorithm [10]. It makes use of external biomedical ontologies and the WordNet as sources of background knowledge [6].

This year, our development of AML was mainly focused on tackling complex matching problems from the new Complex Matching track. Alas, just extending AML to handle the complex EDOAL alignment format took up most of our development time. When we were finally able to start developing matching algorithms, it became clear that each of the numerous types of EDOAL mappings would require its own specialized algorithm, and were only able to develop algorithms for some of the simplest cases, found in the Conference dataset.

We were also unable to fully integrate the code for complex matching with the main AML code-base before the OAEI deadline, and thus participated in the Complex Matching track using a different version of AML, AMLC. In addition to this version and the main AML SEALS version, we participated in the SPIMBENCH and Link Discovery tracks via the HOBBIT platform. In the case of SPIMBENCH, we participated with the

HOBBIT adaptation of the main AML code-base. In the case of Link Discovery, we participated with two specialized versions of AML (AML-Spatial and AML-Linking for the Spatial and Linking tasks respectively) as had been the case in OAEI 2017, due to the unique characteristics of these matching tasks and to the unavailability of the TBox assertions in the HOBBIT datasets.

1.2 Specific Techniques Used

This section describes only the features of AML that are new for the OAEI 2018. For further information on AML's matching strategy, we direct the reader to AML's original paper [7] as well as to the OAEI results publications of the last three editions [4, 5, 3].

1.2.1 Complex AML

For the complex matching track, we focused on the challenge based on the conference ontologies. We developed strategies to identify Attribute Occurrence Restrictions and Attribute Domain Restrictions based on patterns similar to [9]. Attribute Occurrence Restrictions were detected by (1) computing the lexical similarities between the source class and the domains/ranges (or superclasses of domains/ranges) of target properties; (2) selecting target properties with domain/range similarity above a given threshold; (3) building a complex mapping with a comparator and a non-negative integer for the properties with similar domain, adding an inverse property restriction for those with similar range.

Attribute Domain Restrictions were discovered by (1) measuring the lexical similarity between the source class and target classes and selecting target classes above a threshold; (2) removing the matched words from source labels; (3) matching the remaining source strings to target properties and selecting target properties above a threshold; (4) composing a complex mapping which is given a score weighted by the two partial similarities (class and property); (5) selecting complex mappings with scores above a threshold.

1.2.2 Main AML

We made only a few minor changes to the main AML code-base for this OAEI edition.

Instance Matching

In previous OAEI editions, AML's matching strategy for instance matching relied only on Data Property values of individuals and on the relations between individuals. This year, due to the new Knowledge Graph track in which individual matching is expected to be mainly based on their annotations, AML added to its instance matching arsenal the same lexical-based strategy it was already using for class and property matching. However, due to problems in parsing the datasets with the OWL API before the OAEI deadline, we were unable to properly configure this matching strategy and ensure its efficiency.

Interactive Matching

We fixed a bug in AML's interaction manager that was causing it to forget user feedback between the selection and repair steps and thus repeat some questions.

1.3 Adaptations made for the evaluation

As was the case last year, the Link Discovery submissions of AML are adapted to these particular tasks and datasets, as their specificities (namely the absence of a Tbox) demand a dedicated submission. The same is also true to some extent of AML's Complex Matching submission.
As usual, our submission included precomputed dictionaries with translations, to circumvent 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.

2 Results

2.1 Anatomy

AML's result was virtually identical to last year's, with 95% precision, 93.6% recall, 94.3% F-measure, and 83.2% recall++. It was the best ranking system in this track by F-measure.

2.2 Conference

AML's result was exactly the same as last year's, with 74% F-measure according to the full reference alignment 1, 70% F-measure according to the extended reference alignment 2, 78% F-measure according to the discrete uncertain reference alignment, and 77% according to the continuous one. It was the best ranking system in this track or tied for best by F-measure according to 4 of the 5 sets of reference alignments available.

2.3 Multifarm

AML's result was the same as last year's, with 46% F-measure when matching different ontologies and 27% when matching same ontologies. AML was the best ranking system in this track by F-measure in the different ontologies modality.
It is noteworthy that the performance in the same ontologies modality is worse than in the different ontologies, given that the opposite is expected, and indeed was the case for AML prior to 2016. We are unsure as to what led to this relative drop in performance and will have to investigate the matter further.

2.4 Complex Matching

AMLC was configured only for the Conference dataset, in which it obtained 54% precision, 25% recall, and 34% F-measure. It was the only system to produce complex (EDOAL) alignments in this track.

2.5 Interactive Matching

AML had a similar performance to last year's, except that fixing the bug in its interaction manager has prevented repeated queries. We did not yet improve our interactive manager to make feedback requests with sets of conflicting mappings, which would enable us to reduce the total number of user requests AML makes. Thus, the increase in F-measure per user request is relatively low for AML, even if the increase per individual mapping asked is not. It was the system least affected by user errors for the Anatomy dataset, but was substantially more affected than LogMap in the case of the Conference dataset.

2.6 Large Biomedical Ontologies

AML's results were virtually the same as last year's in this track, with an F-measure of 93.3% in FMA-NCI small, 85.5% in FMA-NCI whole, 83.5% in FMA-SNOMED small, 77.2% in FMA-SNOMED whole, 80.1% in SNOMED-NCI small and 76.8% in SNOMED-NCI whole. It was the highest ranked system by F-measure in the FMA-NCI and SNOMED-NCI problems, and the second-highest in the FMA-SNOMED problems.

2.7 Disease and Phenotype

AML generated 2010 mappings in the HP-MP task, 279 of which were unique. It ranked second by F-measure according to the 3-vote silver standard, with 85.6%. In the DOID-ORDO task, it generated by far the most mappings (4749) and the most unique mappings (1886), and as a result had a relatively low F-measure according to the 3-vote silver standard (63.6%).

2.8 Biodiversity and Ecology

AML obtained 86% F-measure in the FLOPO-PTO task and 84.4% F-measure in the ENVO-SWEET task. It ranked first by F-measure in both tasks of this new track.

2.9 SPIMBENCH

AML obtained an F-measure of 86%, ranking second by F-measure. This performance was significantly lower than last year's (92.2%), which was unexpected. We are unsure of whether this is due to a difference in the dataset.

2.10 IIMB

AML obtained a global F-measure of 82.8% across the 20 tasks of this new track, ranking second in F-measure behind LogMap (the only other participating system). It outperformed LogMap in 4 of the tasks, but had a mediocre performance in 6 others.

2.11 Link Discovery

Like in 2017, AML produced a perfect result (100% F-measure) in the Linking and all the Spatial tasks.

2.12 Knowledge Graph

Due to our inability to configure AML's new instance matching strategy prior to the OAEI deadline due to the issues with parsing the datasets for this track, AML took a substantial amount of time to run these datasets, and was unable to finish all of them before the deadline for this manuscript. Nevertheless, for the tasks in which it did complete, it had a high performance in class matching (87% F-measure) but a relatively poor performance in instance matching (28% F-measure).

3 General comments

3.1 Comments on the results

This year, AML was the only system to rise to the challenge of tackling complex ontology matchings, and was the only system to participate in all the tracks. It remained among the highest ranked systems in most of the tracks in which it participated and among the most efficient. The few exceptions to AML's superiority were caused by our inability to test the datasets before the OAEI deadline. We expect to address the remaining challenges in the near future.

3.2 Comments on the OAEI test cases

As always, we welcome the addition of new tracks to the OAEI, and laud the efforts of their organizers, as the effort involved in organizing said tracks cannot be overstated. Nevertheless, we must comment on some of the issues encountered during this OAEI edition, and suggest improvements for the future.

In the new Complex Matching track, we found that the tasks were indeed extremely complex, and in many cases virtually impossible to tackle automatically, as there was insufficient information in the ontologies to derive the type of mappings that were expected. We will work with the organizers to make the tasks more realistic for future OAEI editions, namely by including instance data in the datasets when available.

In the new Knowledge Graph track, the fact that the datasets were not valid OWL (and thus not parsable with the OWL API) before the OAEI deadline was a substantial issue

which prevented us (and undoubtedly other participants) from adequately developing our matching system. For future OAEI editions, we suggest that track organizers test their datasets using a few of the recurring OAEI participating systems.
In the Link Discovery track, we stress once more the need to incorporate TBox information into the datasets so as to enable them to be interpreted automatically without the need for a dedicated parser.
Last but not least, we remain critic of the evaluation in the Disease and Phenotype track by means of silver standards generated from the alignments produced by the participating systems via voting. While we understand that the effort behind building a manually curated reference alignment can be daunting, the current evaluation strategy is unreliable and biased, penalizing systems that are able to find unique mappings that may well be correct. We would welcome an effort to produce a manually curated reference alignments using the silver standards as a starting point.

4 Conclusion

Like in 2017, this year AML was the only matching system to participate in all OAEI tracks, and was among the top performing systems in most of them. AML's performance did not improve in any of the recurring OAEI tracks, as most of our development effort went into tackling new challenges and extending the range of AML. This year, our effort to tackle complex matchings was not well rewarded, as, despite being the only system to generate complex mappings, AML was only able to cover a few of the simplest types of complex mappings. It has become evident that generating such mappings automatically is an extremely difficult task, which requires more effort than that we could devote at this time. Thus, we will continue to address this aspect of ontology matching in the near future.

Acknowledgments

DF was funded by the EC H2020 grant 676559 ELIXIR-EXCELERATE. CP and FMC were funded by the Portuguese FCT through the LASIGE Strategic Project (UID/CEC/00408/2013). FMC was also funded by PTDC/CCI-BIO/28685/2017. CP was also funded by FCT (PTDC/EEI-ESS/4633/2014). The research of IFC and BSB was partially funded by NSF awards CNS-1646395, III-1618126, CCF-1331800, and III-1213013, by NIGMS-NIH award R01GM125943, and by a Bill & Melinda Gates Foundation Grand Challenges Explorations grant.

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. D. Faria, B. S. Balasubramani, V. R. Shivaprabhu, I. Mott, C. Pesquita, F. M. Couto, and I. F. Cruz. Results of AML in OAEI 2017. In *OM-2017: Proceedings of the Twelfth International Workshop on Ontology Matching*, page 122, 2017.
4. D. Faria, C. Martins, A. Nanavaty, D. Oliveira, B. S. Balasubramani, A. Taheri, C. Pesquita, F. M. Couto, and I. F. Cruz. AML results for OAEI 2015. In *Ontology Matching Workshop*. CEUR, 2015.
5. D. Faria, C. Pesquita, B. S. Balasubramani, C. Martins, J. Cardoso, H. Curado, F. M. Couto, and I. F. Cruz. OAEI 2016 results of AML. In *Ontology Matching Workshop*. CEUR, 2016.
6. D. Faria, C. Pesquita, E. Santos, I. F. Cruz, and F. M. Couto. Automatic Background Knowledge Selection for Matching Biomedical Ontologies. *PLoS One*, 9(11):e111226, 2014.
7. 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.
8. 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.
9. D. Ritze, C. Meilicke, O. Šváb Zamazal, and H. Stuckenschmidt. A pattern-based ontology matching approach for detecting complex correspondences. In *Proceedings of the 4th International Conference on Ontology Matching*, volume 551 of *OM'09*, pages 25–36, Aachen, Germany, Germany, 2009. CEUR-WS.org.
10. E. Santos, D. Faria, C. Pesquita, and F. M. Couto. Ontology alignment repair through modularization and confidence-based heuristics. *PLoS ONE*, 10(12):e0144807, 2015.

ALOD2Vec Matcher*

Jan Portisch[0000−0001−5420−0663] and Heiko Paulheim[0000−0003−4386−8195]

Data and Web Science Group, University of Mannheim, Germany
`jan.portisch@sap.com`, `heiko@informatik.uni-mannheim.de`

Abstract. In this paper, we introduce the *ALOD2Vec Matcher*, an ontology matching tool that exploits a Web-scale data set, i.e., *WebIsALOD*, as external knowledge source. In order to make use of the data set, the *RDF2Vec* approach is chosen to derive embeddings for each concept available in the data set.

We show that it is possible to use very large RDF graphs as external background knowledge source for the task of ontology matching.

Keywords: Ontology Matching · Ontology Alignment · External Resources · Vector Space Embeddings · RDF2Vec

1 Presentation of the System

1.1 State, purpose, general statement

The *ALOD2Vec Matcher* is an element-level, label-based matcher which uses a large-scale Web-crawled RDF data set of hypernymy relations as background knowledge. One advantage of that data set is the inclusion of many tail-entities, as well as instance data, such as persons or places, which cannot be found in thesauri. In order to make use of the external data set, a neural language model approach is used to calculate an embedding vector for each concept contained in it.

Given two entities e_1 and e_2, the matcher uses their textual labels to link them to concepts e'_1 and e'_2 in the external data set. Then, the pre-calculated embedding vectors $v_{e'_1}$ and $v_{e'_2}$ of the linked concepts (e'_1 and e'_2) are retrieved and the cosine similarity between those is calculated. Hence: $sim(e_1, e_2) = sim_{cosine}(v_{e'_1}, v_{e'_2})$. The resulting alignment is homogenous, i.e., classes, object properties, and datatype properties are handled separately. In addition, the matcher enforces a one-to-many matching restriction.

1.2 Specific techniques used

For the alignment process, the matcher retrieves textual descriptions of all elements of the ontologies to be matched. A filter adds all simple string matches to the final alignment in order to increase the performance. The remaining labels are linked to concepts of the background data set, are compared, and the best solution is added to the final alignment. A high-level view of the system is depicted in figure 1.

---

* Supported by SAP SE.

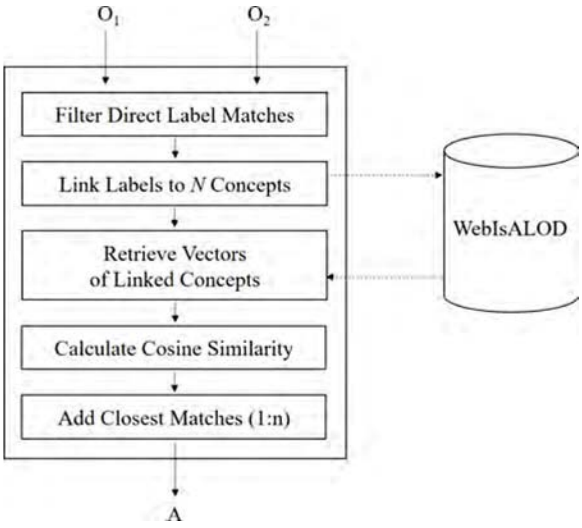


Fig. 1. ALOD2Vec Matching Process

WebIsALOD Data Set When working with knowledge bases in order to exploit the contained knowledge in applications, a frequent problem is the fact that less common entities are not contained within the knowledge base. The *WebIsA* [7] database is an attempt to tackle this problem by providing a data set which is not based on a single source of knowledge – like DBpedia [3] – but instead on the whole Web: The data set consists of hypernymy relations extracted from the *Common Crawl*[1], a freely downloadable crawl of a significant portion of the Web. A sample triple from the data set is *european_union skos:broader international_organization*[2]. The data set is also available via a Linked Open Data (LOD) endpoint[3] under the name *WebIsALOD* [2]. In the LOD data set, a machine-learned confidence score $c \in [0, 1]$ is assigned to every hypernymy triple indicating the assumed degree of truth of the statement.

RDF2Vec The background data set can be viewed as a very large knowledge graph; in order to obtain a similarity score for nodes in that graph, the *RDF2Vec* [6] approach is used. It applies the *word2vec* [4,5] model to RDF data: Random walks are performed for each node and are interpreted as sentences. After the walk generation, the sentences are used as input for the word2vec algorithm. As a result, one obtains a vector for each word, i.e., a concept in the RDF graph. The approach is used here to obtain vectors for all concepts in the WebIsALOD data set.

---

[1] see `http://commoncrawl.org/`

[2] see `http://webisa.webdatacommons.org/concept/european_union_`

[3] see `http://webisa.webdatacommons.org/`

Linking The first step is to link the obtained labels from the ontology to concepts in the WebIsALOD data set. Therefore, string operations are performed on the label and it is checked whether the label is available in WebIsALOD. If it cannot be found, labels consisting of multiple words are truncated from the right, and the process is repeated to check for sub-concepts. For example, the label *United Nations Peacekeeping Mission in Mali* cannot be found in WebIsALOD. Therefore, it is truncated until the longest label from the left is found – in this case *United Nations*. The process is repeated until all tokens are processed. The resulting concepts for the given label are: *United Nations*[4], *peacekeeping mission*[5], and *Mali*[6].

Similarity Calculation As stated before, labels are linked to concepts, their vectors are retrieved, and the cosine similarity between them is used as similarity score.

There are cases in which parts of a label cannot be found, however, for example in *tubule macula* and in *macula lutea* both times only *macula* can be found using the WebIsALOD data set. If only the found concepts would be used to calculate the similarity between the concepts, a perfect score would be obtained because $sim(macula, macula) = 1.0$. This is not precise as the approach does not allow to discriminate between perfect matches due to incomplete linking and *real* perfect matches. Therefore, a penalty factor $p \in [0, 1]$ is introduced that is to be multiplied with the final similarity score and which lowers the score for incomplete links; $p = 0$ indicates the maximal penalty, $p = 1$ indicates no penalty. The calculation of p is depicted in equation 1:

$$p = 0.5 * \frac{|Found\ Concepts\ L_1|}{|Possible\ Concepts\ L_1|} + 0.5 * \frac{|Found\ Concepts\ L_2|}{|Possible\ Concepts\ L_2|} \tag{1}$$

where L_1 is the label of the first concept and L_2 is the label of the second one; $|Found\ Concepts\ L_i|$ is the number of tokens for which a concept could be found (minus stopwords) and $|Possible\ Concepts\ L_i|$ is the number of tokens of the label without stopwords. The penalty score is multiplied with the final similarity score. Hence, incomplete linkages are penalized.

If two labels were matched to multiple concepts, a resolution is required. In this case the best average similarity is used:

$$sim_{average} = \frac{\Sigma_{i \in c_1}^{|c_1|} Max_{j \in c_2}^{|c_2|} sim(c_{1_i}, c_{2_j})}{|c_1|} \tag{2}$$

where c_1 and c_2 represent two individual concepts and c_{1_i}, respectively c_{2_j}, represent the i^{th} and j^{th} sub-concept of c_1 and c_2; $|c_1|$ and $|c_2|$ are the number of sub-concepts of c_1 and c_2; c_1 is the concept with more tokens.

---

[4] see `http://webisa.webdatacommons.org/concept/united_nations_`

[5] see `http://webisa.webdatacommons.org/concept/peacekeeping_mission_`

[6] see `http://webisa.webdatacommons.org/concept/_mali_`

Typically, there is more than one label to an entity of an ontology. Therefore, a score-matrix is used: Every label of an entity is linked and compared to every label of the other entity and the best score is returned.

RDF2Vec Configuration Parameters We generated 100 sentences of depth 8 for each node in the WebIsALOD data set for the training process of the model. In order to have also sentences for nodes that do not have out-going edges, those were identified and sentences were generated backwards and afterwards reversed. The sentences were generated in a biased fashion [1], i.e., high-confidence edges are followed with a higher probability. Eventually, the embeddings were trained using the continuous bag of words (CBOW) approach with the parameters of the original RDF2Vec paper: $window\ size = 5$, $number\ of\ iterations = 5$, $negative\ sampling = true$, $negative\ samples = 25$, $average\ input\ vector = true$, and 200 dimensional embeddings.

2 Results

2.1 Anatomy

For the Anatomy data set, the matcher achieves a higher recall and F_1 score compared to the baseline solution. However, the true positives are mostly exact lexical matches or share many common tokens.
Concerning runtime-performance, *ALOD2Vec Matcher* performs in the upper half of all matchers that participated in the Anatomy track.

2.2 Conference

On the Conference data set, it can be seen that the matcher is better in aligning classes than in aligning properties. This is in line with the results reported for other matchers. In this case, it is due to fewer lexical matches in properties as well as the higher usage of non-nouns which cannot be properly linked to the background knowledge source.

2.3 Large BioMed

For the Large BioMed matching tasks, the matcher is capable of aligning the small fragments within the given time frame of 6 hours. While *ALOD2Vec Matcher* performs slightly above the 2017 and 2018 F_1 averages on the small FMA-NCI data set, it perfoms in the lower half for the remaining ones.

2.4 Complex Track

Although the matcher presented here is not capable of generating complex correspondences yet, it could produce results for the entity identification subtask for two data sets: On GeoLink, *ALOD2Vec Matcher* achieved the highest F_1 score and recall of all matchers that participated; on Hydrograph, alignments for the English ontologies could be generated and scored within the median.

3 General Comments

3.1 Comments on the results

The matcher performs above the given baselines. However, the matches are still rather trivial and mostly share common tokens.

There are multiple reasons for the mediocre performance. First, the underlying data set is very noisy: It contains a lot of wrong information (e.g. *fish skos:broader fisher*)[7], subjective information (e.g. *donald_trump skos:broader lunatic*)[8], and is not strictly hierarchical (e.g. *live skos:broader quality*, and vice versa)[9]. In addition, the tail-entity problem is still not solved because very specific entities are involved in very few hypernymy statements and their resulting vectors are likely not meaningful (e.g. *complex congenital heart defect*)[10].

Besides the pitfalls of the data set, the matcher cannot handle homonyms, non-nouns, or non-English labels.

3.2 Discussions on the way to improve the proposed system

There are three ways in which the current research focusing on this approach can be improved in the future: Firstly, more propositionalization techniques for very large data sets could be explored. Secondly, the matcher itself can be enhanced to use more information available in ontologies such as their structure. And lastly, the data sets to be used can be improved. WebIsALOD is only one Web-scale RDF data set and still has some pitfalls such as the restriction to hypernymy relations and noise. More such data sets can be created and used in the future.

4 Conclusion

In this paper, we presented the *ALOD2Vec Matcher*, a matcher utilizing a Web-crawled knowledge data set by applying the RDF2Vec methodology to a hypernymy data set extracted from the Web. It could be shown that it is possible to use very large RDF graphs as external background knowledge and the RDF2Vec methodology for the task of ontology matching.

References

1. Cochez, M., Ristoski, P., Ponzetto, S.P., Paulheim, H.: Biased graph walks for rdf graph embeddings. In: Proceedings of the 7th International Conference on Web Intelligence, Mining and Semantics. p. 21. ACM (2017)

---

[7] see `http://webisa.webdatacommons.org/concept/_fish_`

[8] see `http://webisa.webdatacommons.org/concept/donald_trump_`

[9] see `http://webisa.webdatacommons.org/concept/_quality_`

[10] see `http://webisa.webdatacommons.org/concept/complex+congenital+heart_defect_`

2. Hertling, S., Paulheim, H.: WebIsALOD: Providing Hypernymy Relations Extracted From the Web as Linked Open Data. In: International Semantic Web Conference. pp. 111–119. Springer (2017)
3. Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P.N., Hellmann, S., Morsey, M., Bizer, C.: DBpedia A Large-scale, Multilingual Knowledge Base Extracted from Wikipedia. Semantic Web **6**(2), 167–195 (2012)
4. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient Estimation of Word Representations in Vector Space. arXiv preprint arXiv:1301.3781 (2013)
5. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed Representations of Words and Phrases and Their Compositionality. In: Advances in Neural Information Processing Systems. pp. 3111–3119 (2013), `http://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality`
6. Ristoski, P., Rosati, J., Di Noia, T., De Leone, R., Paulheim, H.: RDF2vec: RDF Graph Embeddings and Their Applications. Semantic Web Journal (2017), `http://www.semantic-web-journal.net/system/files/swj1495.pdf`
7. Seitner, J., Bizer, C., Eckert, K., Faralli, S., Meusel, R., Paulheim, H., Ponzetto, S.P.: A Large DataBase of Hypernymy Relations Extracted from the Web. In: LREC (2016)

CANARD complex matching system: results of the 2018 OAEI evaluation campaign

Elodie Thiéblin, Ollivier Haemmerlé, Cassia Trojahn

IRIT & Université de Toulouse 2 Jean Jaurès, Toulouse, France
`{firstname.lastname}@irit.fr`

Abstract. This paper presents the results obtained by the CANARD system in the OAEI 2018 campaign. CANARD can produce complex alignments. This is the first participation of CANARD in the campaign. Even though the system has been able to generate alignments for one only complex dataset (Taxon), the results are promising.

1 Presentation of the system

1.1 State, purpose, general statement

The CANARD (Complex Alignment Need and A-box based Relation Discovery) system discovers complex correspondences between populated ontologies based on Competency Questions for Alignment (CQAs). Competency Questions for Alignment (CQAs) represent the knowledge needs of a user and define the scope of the alignment [3]. They are competency questions that need to be satisfied over two or more ontologies. Our approach takes as input a set of CQAs translated into SPARQL queries over the source ontology. The answer to each query is a set of instances retrieved from a knowledge base described by the source ontology. These instances are matched with those of a knowledge base described by the target ontology. The generation of the correspondence is performed by matching the graph-pattern from the source query to the lexically similar surroundings of the target instances.

1.2 Specific techniques used

The CQAs that are taken as input by CANARD are limited to class expressions (interpreted as a set of instances). The approach is developed in 11 steps, as depicted in Figure 1:

① Extract source DL formula e_s from SPARQL CQA.
② Extract lexical information from the CQA, L_s set labels of atoms from the DL formula.
③ Extract source instances $inst_s$.
④ Find equivalent or similar (same label) target instances $inst_t$ to the source instances $inst_s$.

⑤ Retrieve description of target instances: set of triples and object/subject type.

⑥ For each triple, retrieve L_t labels of entities.

⑦ Compare L_s and L_t using a string comparison metric (e.g., Levenshtein distance with a threshold).

⑧ Keep the triples with the summed similarity of their labels above a threshold τ. Keep the object(/subject) type if its similarity is better than the one of the object(/subject).

⑨ Express the triple into a DL formula e_t.

⑩ Aggregate the formulae e_t into an explicit or implicit form: if two DL formulae have a common atom in their right member (target member), the atoms which differed are put together.

⑪ Put e_s and e_t together in a correspondence ($e_s \equiv e_t$) and express this correspondence in EDOAL. The average string similarity between the aggregated formula and the CQA labels gives the confidence value of the correspondence.

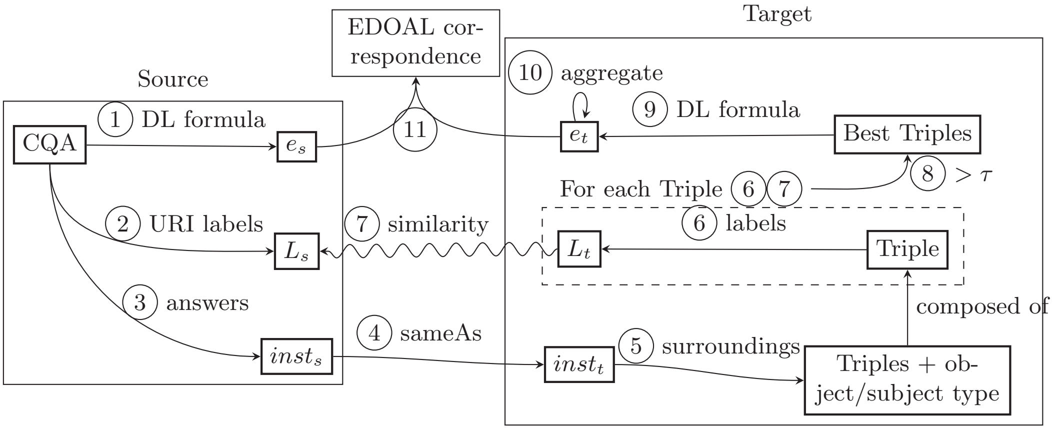


Fig. 1: Schema of the general approach.

The instance matching phase (step ④) is based on existing *owl:sameAs*, *skos:closeMatch*, *skos:exactMatch* and exact label matching. The similarity between the sets of labels L_s and L_t of step ⑦ is the cartesian product of the string similarities between the labels of L_s and L_t (equation 1).

$$sim(L_s, L_t) = \sum_{l_s \in L_s} \sum_{l_t \in L_t} strSim(l_s, l_t) \tag{1}$$

$strSim$ is the string similarity between two labels l_s and l_t (equation 2). τ is the threshold for the similarity measure. In our experiments, we have empirically set up $\tau = 0.5$.

$$strSim(l_s, l_t) = \begin{cases} \sigma \text{ if } \sigma > \tau, \text{ where } \sigma = 1 - \dfrac{levenshteinDist(l_s, l_t)}{\max(|l_s|, |l_t|)} \\ 0 \text{ otherwise} \end{cases} \tag{2}$$

The confidence value given to the final correspondence (step ⑪) is the similarity of the triple it comes from or average similarity if it comes from more than one triple. The confidence value is reduced to 1 if it is initially calculated over 1.

1.3 Adaptations made for the evaluation

Automatic generation of CQAs The CQAs can not be given as input in the evaluation as none are available in the OAEI datasets. We developed a CQA generator that was integrated to the version of the system used in the evaluation. This generator produces two types of SPARQL queries: *Classes* and *Property-Value pairs*.

Classes For each *owl:Class* populated with at least one instance, a SPARQL query is created to retrieve all the instances of this class. If `<o1#class1>` is a populated class of the source ontology, the following query is created:
`SELECT DISTINCT ?x WHERE {?x a <o1#class1>.}`

Property-Value pairs Inspired by the approaches of [1,2,4], we create SPARQL queries of the form

- `SELECT DISTINCT ?x WHERE {?x <o1#property1> <o1#Value1>.}`
- `SELECT DISTINCT ?x WHERE {<o1#Value1> <o1#property1> ?x.}`
- `SELECT DISTINCT ?x WHERE {?x <o1#property1> "Value".}`

These property-value pairs are computed as follow: for each property (object or data property), the number of distinct object and subject values are retrieved. If the ratio of these two numbers is over a threshold (arbitrarily set to 30) and the smallest number is smaller than a threshold (arbitrarily set to 20), a query is created for each of the less than 20 values. For example, if the property `<o1#property1>` has 300 different subject values and 3 different object values (`"Value1"`, `"Value2"`, `"Value3"`), the ratio $|subject|/|object| = 300/3 > 30$ and $|object| = 3 < 20$. The 3 following queries are created as CQAs:

- `SELECT DISTINCT ?x WHERE {?x <o1#property1> "Value1".}`
- `SELECT DISTINCT ?x WHERE {?x <o1#property1> "Value2".}`
- `SELECT DISTINCT ?x WHERE {?x <o1#property1> "Value3".}`

The threshold on the smallest number ensures that the property-value pairs represent a category. The threshold on the ratio ensures that properties represent categories and not properties with few instanciations.

Implementation adaptations In the initial version of the system, Fuseki server endpoints are given as input. For the SEALS evaluation, we embedded a Fuseki server inside the matcher. The ontologies are downloaded from the SEALS repository, then uploaded in the embedded Fuseki server before the matching process can start. This downloading-uploading phase may take time, in particular when dealing with large files.

The CANARD system in the SEALS package is available at `http://doi.org/10.6084/m9.figshare.7159760.v1`. The generated alignments in EDOAL format are available at `http://oaei.ontologymatching.org/2018/results/complex/taxon/CANARD.html` (link to each pair of task). Note that, as described below, CANARD was able to generate results for the Taxon track.

2 Results

The CANARD system could only output correspondences for the Taxon dataset of the Complex track. Indeed, the other datasets of this track do not contain instances and least of all common instances.

Table 1 shows the run-time of CANARD on all pairs of ontologies in the Taxon track, as well as the characteristics of the output alignments. As the alignment process is directional, we do not obtain symmetrical results for a pair of ontologies. CANARD is able to generate different kinds of correspondences: (1:1), (1:n) and (m:n). The best precision was obtained for the pair agronomicTaxon-agrovoc with a precision of 0.57. CANARD did not output any correspondence for 4 oriented pairs (in grey in Table 1). These empty results can be due to the fail of the instance matching phase of our approach. We could observe that with TaxRef as the source knowledge base, no correspondence could be generated. The exception is the pair *taxref-agrovoc* where 8 correspondences were found but only involving *skos:exactMatch* or *skos:closeMatch* properties in the constructions. The incorrect correspondences of this pair have a low confidence (between 0.05 and 0.30).

Looking for the query rewriting task in Taxon, CANARD's alignment was used to rewrite the most queries (best *qwr*). As CANARD does not deal with binary CQAs, none of the 3 binary queries × 12 pairs of ontologies = 36 binary query cases could be dealt with. Out of the 2 unary queries × 12 pairs = 24 unary query cases, CANARD could deal with 6 unary cases needing a complex correspondence and 2 needing simple correspondences for a total of (8/24) 33% of unary query cases.

Overall, for the query cases needing complex correspondences, (0+6/28+16) 14% were covered by CANARD. For all the query cases, the CANARD system could provide an answer to (8/36+24) 13% of all cases.

3 General comments

The CANARD approach relies on common instances between the ontologies to be aligned. Hence, when such instances are not available, as for the Conference,

Test Case ID	Run Time (s)	output corres.	correct corres.	prec.	(1:1)	(1:n)	(m:n)
agronomicTaxon-agrovoc	37	7	4	0.57	0	7	0
agronomicTaxon-dbpedia	75	17	3	0.18	3	14	0
agronomicTaxon-taxref	87	9	3	0.33	1	8	0
agrovoc-agronomicTaxon	20	0		NaN	0	0	0
agrovoc-dbpedia	128	13	3	0.23	0	0	13
agrovoc-taxref	87	8	0	0	0	0	8
dbpedia-agronomicTaxon	556	0		NaN	0	0	0
dbpedia-agrovoc	236	37	0	0	0	20	17
dbpedia-taxref	333	43	14	0.33	0	17	26
taxref-agronomicTaxon	269			NaN	0	0	0
taxref-agrovoc	283	8	0	0	0	0	8
taxref-dbpedia	351	0		NaN	0	0	0
Global	2468	142	27	0.20	4	66	72

Table 1: Results of CANARD on the Taxon track

GeoLink and Hydrography datasets, the approach is not able to generated complex correspondences. Furthermore, CANARD is need-oriented and requires a set competency questions to guide the matching process. Here, these "questions" have been automatically generated based on a set of patterns.

The current version of the system is limited to finding complex correspondences involving classes and properties are not yet taken into account. We plan to extend the systems to take binary relations in the next version. Another point that we would like to improve is the semantics of the confidence of the correspondences.

With respect to the technical environment, as mentioned before, the initial version of the system receives as input the endpoints of the populated ontologies. Using SEALS, the large ontologies are stored into repositories. Our systems hence downloads them and stores them into an embedded Fuseki server. This configuration is not ideal as we have to deal with large knowledge bases. Furthermore, we struggled with the SEALS dependencies in order to correctly package our system into the SEALS format.

As we focus on user needs in order to avoid dealing with the whole alignment space, it could be interesting to having more need-oriented tasks with respect to the alignments coverage.

4 Conclusions

This paper presented the adapted version of the CANARD system and its preliminary results in the OAEI 2018 campaign. This year, we have been participated only in the Taxon track, in which ontologies are populated with common instances. CANARD was the only system to output complex correspondences on the Taxon track.

Acknowledgements

Cassia Trojahn has been partially supported by the French CIMI Labex projet IBLiD (Integration of Big and Linked Data for On-Line Analytics).

References

1. Parundekar, R., Knoblock, C.A., Ambite, J.L.: Linking and building ontologies of linked data. In: ISWC. pp. 598–614. Springer (2010)
2. Parundekar, R., Knoblock, C.A., Ambite, J.L.: Discovering concept coverings in ontologies of linked data sources. In: ISWC. pp. 427–443. Springer (2012)
3. Thiéblin, E., Haemmerlé, O., Trojahn, C.: Complex matching based on competency questions for alignment: a first sketch. In: Ontology Matching Workshop. p. 5 (2018)
4. Walshe, B., Brennan, R., O'Sullivan, D.: Bayes-recce: A bayesian model for detecting restriction class correspondences in linked open data knowledge bases. International Journal on Semantic Web and Information Systems 12(2), 25–52 (2016)

DOME results for OAEI 2018

Sven Hertling and Heiko Paulheim

Data and Web Science Group, University of Mannheim, Germany
`{sven,heiko}@informatik.uni-mannheim.de`

Abstract. DOME (Deep Ontology MatchEr) is a scalable matcher which relies on large texts describing the ontological concepts. Using the doc2vec approach, these texts are used to train a fixed-length vector representation of the concepts. Mappings are generated if two concepts are close to each other in the resulting vector space. If no large texts are available, DOME falls back to a string based matching technique. Due to its high scalability, it can also produce results in the largebio track of OAEI and can be applied to very large ontologies. The results look promising if huge texts are available, but there is still a lot of room for improvement.

1 Presentation of the system

1.1 State, purpose, general statement

Ontology matching is often based on string comparisons because each resource is described by URI fragments (last part of an URI after the # sign), `rdfs:label`s, and `rdfs:comment`s. The DOME matcher specifically relies on large texts which describes the resources, and thereby allows to make a better distinction in case of a similar labels. Especially in knowledge graphs like DBpedia or YAGO, such texts are easily extracted from the corresponding Wikipedia abstract.

The usual problem with such large texts is the matching with other similar and long texts. One possible way is to use topic modeling like latent semantic analysis(LSA [2]) or latent dirichlet allocation (LDA [1]). The extracted topics can then be used to find overlaps and in the end similar concepts.

DOME uses another approach called doc2vec (also paragraph vector [5]) which is based on word2vec [6]. The idea is to represent a variable-length texts, like sentences, paragraphs, and documents, as a fixed-length feature vector. This vector is trained to predict the words appearing in the document. Thus this vector represents the semantics of the concept when training on texts which defines the meaning of the concept.

Two approaches for training this vector are established: Distributed Memory (DM) and Distributed Bag of Words (DBOW). Applied to an example concept like Harry Potter[1] the framework of DM is shown in figure 1. During training, the algorithm iterates over the given text in a sliding window of a specified and fixed length. The goal is to predict the last word given the first n words. One

---

[1] `http://harrypotter.wikia.com/wiki/Harry_Potter`

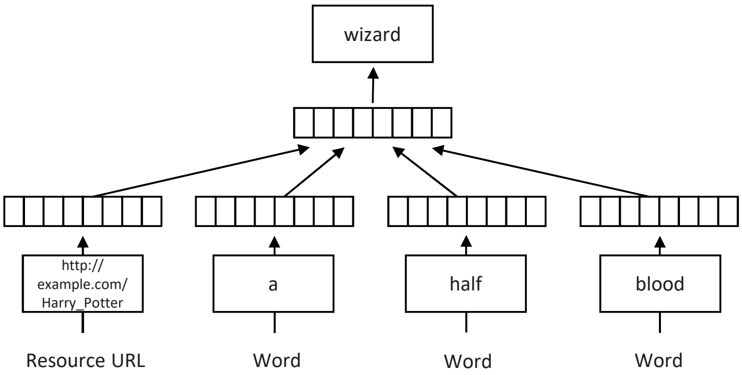


Fig. 1. Training of Distributed Memory given the concept Harry Potter and a small excerpt of the corresponding wiki abstract.

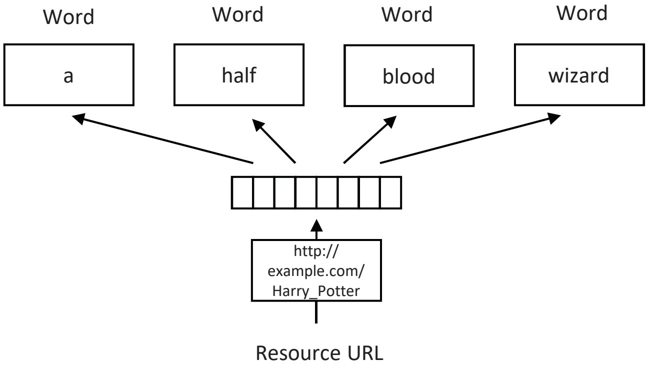


Fig. 2. Training of Distributed Bag of Words. The example is the same as in figure 1 but now the concept URI together with a small subset of text is usbed to predict the following word.

special vector is the first one which represents the paragraph vector. In our case this is the URI of the concept. All large texts which define this resource can be used to train this vector.

Another approach for generating the concept vector is Distributed Bag of Words (DBOW), shown in figure 2. Instead of using concept vectors for each word, it tries to predict words from the text as an output.

DOME uses the DM sequence learning algorithm with a vector size of 300 and window size of 5. The training is repeated in 10 epochs. The minimal word frequency is set to the minimum to allow all words contribute to the concept vector. We compute a predefined set of properties which contains definitional texts by two simple rules: 1) directly choose `rdfs:comment` 2) use every property where the URI ends in “abstract”. This can be further improved in the next version of DOME.

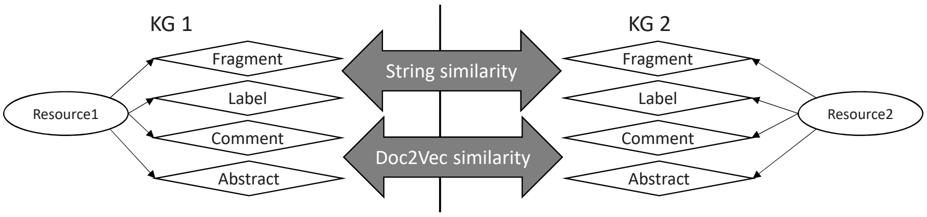


Fig. 3. Matching strategy of DOME

The doc2vec model is trained on all texts available in both ontologies. For each concept in the second ontology, the corresponding concept vector is computed, and the concepts which have the most similar vectors to those from the first ontology are retrieved. A mapping between two resource is established when the cosine similarity is above 0.85 (the threshold has been chosen based on a manual inspection of the results).

The whole matching approach is shown in figure 3. The labels and fragments of each resource are compared using string based similarity. Specifically, the texts are tokenized and all punctuation (especially underscores and the like) are removed. After lowercasing, these values are stored in a hash structure. A mapping is created when each fragment or label have an exact match. The confidence value of these alignments are set to 1.0. After this step, the doc2vec approach is applied to find further matching concepts. We ensured that the mapping is OWL compliant because we only match instances to instances, classes to classes, and properties to properties. In the latter case we further distinguish datatype properties and object properties but also match properties declared as rdf:properties. With such a setup, the matcher is very scalable and can match all types of resources.

1.2 Specific techniques used

The main technique used in DOME is the doc2vec approach [5] for comments and abstracts of concepts. It is only activated when there is enough text to process. All other matching techniques rely on fast string similarity. Further filtering of the alignment is not executed but during the matching only one to one mappings are allowed.

1.3 Adaptations made for the evaluation

DOME is implemented in java and uses the DL4J[2] (Deep Learning for Java) as an implementation of the doc2vec approach. DL4J heavily relies on platform specific implementations which are stored in multiple JAR files. This allows it to make use of GPUs to further speed up the computation. DOME relies on the

---

[2] `https://deeplearning4j.org`

CPU implementation of DL4J because upfront it is not clear if all evaluation machines used for OAEI contain a DL4J compatible GPU.

Although the DL4J framework allows for searching for related concepts, it does not provide the similarity values out of the box. Thus the framework is modified to also retrieve these value which can be used in the alignment file to represent the confidence of a mapping. Since the values are already normalized no further post processing step of the similarity values is needed.

Unfortunately, the packaged SEALS matcher was not able to run under the SEALS evaluation routine. The SEALS client loads all JAR files in its own classpath. This is a very secure way of running third-party code, but at the same time one of the most frequent cauess of matchers not working at OAEI, as in the case of DOME. The root cause is the custom classloader of SEALS which uses the JCL library[3]. The SEALS classloader is a subclass of the AbstractClassLoader in the JCL library. Both classloaders do not implement all methods (especially the getPackage method) of the standard classloader. Many other libraries use such functions to further load operating specific code. This applies to the DL4J library as well as the sqlite-jdbc library.

We fixed the error by creating an intermediate matcher which calls another java process. Within that process the classloader is the standard one and the DL4J library could be loaded without any errors. We released a matching framework which does the SEALS and Hobbit packaging, uploading and creating the intermediate matcher.[4]

1.4 Link to the system and parameters file

DOME can be downloaded from
`https://www.dropbox.com/s/1bpektuvcsbk5ph/DOME.zip?dl=0`.

2 Results

The following section discusses the results for each track of the OAEI 2018 where DOME is able to produce meaningful results. This includes the anatomy, conference, largebio, phenotype, and knowledge graph track.

DOME was not able to complete the multifarm track because currently no translation component is included. This would be possible with cross lingual embedding approaches shown in [8]. For complex and interactive track the matching system has to produce different type of output mapping or matching strategy which is not implemented. The tracks biodiv and iimb don't contain enough free texts in the selected properties.

---

[3] `https://github.com/kamranzafar/JCL`

[4] `https://github.com/sven-h/ontMatchingHobbit`

2.1 Anatomy

In the anatomy track, there are only labels given, thus the doc2vec approach is not used here. There are some properties like `oboInOwl:hasRelatedSynonym` or `oboInOwl:hasDefinition` which point to resources with more describing text, but these resources is not recognized by DOME, since we do not implement a larger list of properties used to point to texts.

Therefore, DOME only utilizes string based matching for this track. The text is lowercased, tokenized and then matched based on a hashing algorithm. This results in a high precision of 0.997 (similar to the string equivalence baseline) and a very low runtime of 22 seconds. Only LogMapLt was 4 seconds faster.

Due to a slightly lower recall of 0.615 (0.07 lower than the baseline) DOME has a lower F-Measure than the baseline.

In improvement in this track would be to use the additional texts from `oboInOwl:hasRelatedSynonym` and `oboInOwl:hasDefinition` to further increase the recall. In order not to have to manually maintain such a list, it would also be possible to incorporate all literals that consist of text of at least a certain number of words.

2.2 Conference

Within the conference track, DOME is as bit better than the baseline and often similar to edna (which is a string editing distance matcher adopted from the benchmark track). Evaluating DOME against the original reference alignment it performs exactly like edna in the class mappings and a bit better in the property mappings - both in terms of recall and precision. This results in 0.07 better F-Measure. But there is room for a lot improvement, because in this year, the best matcher reached 0.58 F-Measure in this track.

When comparing to the entailed reference alignment DOME has same evaluation measures like edna and a bit better when comparing properties. If both classes and properties are taken into account DOME is only 0.01 better than edna and 0.15 behind the current best matcher.

In most of the conference ontologies, there are no long natural language texts. Only in rare cases, some classes are described by a comment. Those were processed by the doc2vec model but does not yield any new mappings.

2.3 Largebio

In the largebio track, the number of classes is very high. In the case of FMA-SNOMED this results in matching 78,989 classes to 122,464 classes. Matchers which compare a string from one ontology to all concepts of the other ontology have a quadratic runtime and usually can not finish in time. DOME is one of five matchers (DOME, FCAMapX, LogMap, LogMapBio, XMap) which were able to return results within the given time limit. It is the fastest one and terminates within 30 seconds on the largest track. The second fastest is XMap with 7 minutes and the slowest one is LogMapBio with 49 minutes. The reason here is the same as in the anatomy track. Most resources are only described by a label and fragment without further textual content. Thus, DOME relies on string comparison with a high precision but low recall. In case of “SNOMED-NCI whole”, this results in a precision of 0.907 and a recall of 0.485 (F-Measure of 0.632). The best matcher on this subtrack in terms of F-Measure is FCAMapX with a value of 0.733.

2.4 Phenotype

The phenotype track is based on a real use case, and the matcher should find alignments between disease and phenotype ontologies. DOME is also able to complete this track but with a low F-Measure of 0.483 (HP-MP) and 0.633 (DOID-ORDO). The precision is again the highest among all matchers, but the recall is below 0.5.

However, some ontologies in this track, like the DOID ontology, have further properties containing describing texts like `obo:IAO_0000115` (label of the property is definition). DOME in its current version does not make use of this property, but, as discussed for the anatomy track above, those could be utilized by extending the system.

2.5 Knowledge Graph

The knowledge graph track is a new track where classes, properties and instances should be matched. As already pointed out in [3,4], matching the classes and properties is easier than the instances. This is also the case for the DOME matcher.

It returns all three types of mappings and complete on all nine sub tasks. In average it returns 16 class, 207 property, and 15,912 instance mappings.

DOME achieved an F-Measure of 0.73 in the class correspondences. It is balanced between recall and precision, but even the baseline has a higher recall. So there should some room for improvement.

When analyzing the property alignments, only DOME and the baseline can produce any results. Most likely, the reason is that all properties are typed as `rdf:Property` and not subdivided into `owl:DatatypeProperty` and `owl:ObjectProperty`. As discussed above, DOME is configured to match also `rdf:Property`. This results in a F-Measure of 0.84.

Instance matches are generated by AML, DOME, LogMap, LogMapLt and the baseline. Especially in the instance mapping the doc2vec approach can help because long comments and abstracts of the resources are available. DOME was the second best matcher with an F-Measure of 0.61 (the baseline is the best “matcher” with an F-Measure of 0.69).

Overall, looking at the results for classes, properties, and instances together, DOME has an F-Measure of 0.68, which is better than all matchers except the baseline.

3 General comments

3.1 Comments on the results

The overall results shows that DOME is in a development phase. Sometimes it can beat at least the baselines in terms of F-measure and sometimes not. Currently there are not many tracks which provide a large amount of describing text for each resource, but many ontologies and knowledges graphs exists out there where this is the case.

3.2 Discussions on the way to improve the proposed system

Based on the evaluation on all kinds of different tracks, we noticed a lot of further improvements. First of all, some ontologies use properties which connect a resource to its describing text which are not recognized by DOME. One possible approach to fix this would be the use all properties which contain long texts by some heuristic, e.g., strings exceeding a certain number of characters on average. This would include more text to help the doc2vec model to better differentiate the concepts.

Another possible improvement is to use pretrained word vectors. Those might contain more semantics for each word than training it directly on describing texts for the two ontologies. However, for some very domain-specific ontologies with large amounts of texts, the generic pre-trained embeddings might even perform worse, thus, it is an open research question which of the two yields better results.

A third possible approach is to combine the approach of RDF2Vec [7] (i.e., computing the word2vec embedding of random walks within knowledge graphs) and various cross lingual embeddings shown in [8]. One simple approach would be to learn a linear transformation between the two generated embeddings of the ontologies.

4 Conclusions

In this paper, we have introduced the DOME matcher, which relies on document embeddings for texts describing the concepts defined in an ontology. The results for DOME are analyzed on the different tracks of OAEI. DOME is a highly scale matching system capable of generating class, property and instance alignments. On some tracks where a lot of text describing each resource exists, it shows promising results. However, the matcher is currently in an early state and offers a lot of room for improvement.

References

1. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. Journal of machine Learning research 3(Jan), 993–1022 (2003)
2. Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K., Harshman, R.: Indexing by latent semantic analysis. Journal of the American society for information science 41(6), 391–407 (1990)
3. Hertling, S., Paulheim, H.: Dbkwik: A consolidated knowledge graph from thousands of wikis. In: IEEE International Conference on Big Knowledge, ICBK 2018, Singapore (2018)
4. Hofmann, A., Perchani, S., Portisch, J., Hertling, S., Paulheim, H.: Dbkwik: Towards knowledge graph creation from thousands of wikis. In: Proceedings of the International Semantic Web Conference (Posters and Demos), Vienna, Austria. pp. 21–25 (2017)
5. Le, Q., Mikolov, T.: Distributed representations of sentences and documents. In: International Conference on Machine Learning. pp. 1188–1196 (2014)
6. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems. pp. 3111–3119 (2013)
7. Ristoski, P., Paulheim, H.: Rdf2vec: Rdf graph embeddings for data mining. In: International Semantic Web Conference. pp. 498–514. Springer (2016)
8. Ruder, S., Vulić, I., Søgaard, A.: A survey of cross-lingual word embedding models. arXiv preprint arXiv:1706.04902 (2017)

EVOCROS: Results for OAEI 2018

Juliana Medeiros Destro[1], Gabriel Oliveira dos Santos[1], Julio Cesar dos Reis[1], Ricardo da S. Torres[1], Ariadne Maria B. R. Carvalho[1], and Ivan Luiz Marques Ricarte[2]

[1] Institute of Computing, University of Campinas, Campinas-SP, Brazil
`{juliana.destro,jreis,rtorres,ariadne}@ic.unicamp.br`
`gabriel.santos@students.ic.unicamp.br`
[2] School of Technology, University of Campinas, Limeira-SP, Brazil
`ricarte@unicamp.br`

Abstract. This paper describes EVOCROS, a cross-lingual ontology alignment system suited to create mappings between ontologies described in different natural language. Our tool combines semantic and syntactic similarity measures in a weighted average metric. The semantic is computed via NASARI vectors used together with *BabelNet*, which is a domain-neutral semantic network. The tool employs automatic translation to a pivot language to consider the similarity. EVOCROS was tested and obtained high quality alignment in the Multifarm dataset. We discuss the experimented configurations and the achieved results in OAEI 2018. This is our first participation in OAEI.

Keywords: cross-lingual matching · semantic matching · background knowledge

1 Presentation of the system

There is a growing number of ontologies described in different natural languages. The mappings among different ontologies are relevant for the integration of heterogeneous data sources to facilitate the exchange of information between systems. Although automatic monolingual ontology matching has been extensively investigated [7], cross-lingual ontology matching still demands further investigations aiming to automatically identify correspondences between ontologies described in different languages. EVOCROS is our attempt at automatic cross-lingual ontology matching, inspired from experiments on the influence of syntactic and semantic similarity measures in ontology matching algorithms [1]. In this section, we describe the system and the implemented techniques.

1.1 State, purpose, general statement

EVOCROS is a cross-lingual ontology alignment tool based on a composed similarity measure relying on both syntactic and semantic similarity techniques. Syntactic similarity may be understood as a score calculated based on string analysis (extracted from labels of concepts), whereas the semantic similarity is computed taking into account background knowledge. Our approach computes a weighted mean of semantic and syntactic similarities.

1.2 Specific techniques used

The tool is developed in Python 3. It works by comparing the computed similarity between a concept from an ontology (in its automatically translated version) to another concept from a different ontology. The concept terms are translated to a pivot natural language aiming to use available external resources such as thesauri, corpora, dictionaries, *etc.* to overcome the language and alphabet barriers.

Figure 1 presents the workflow of the tool. The first step is the pre-processing of the source and target input ontologies, converting them into owlready2[3] objects. Each concept of the source ontology is compared to all concepts of the target ontology.

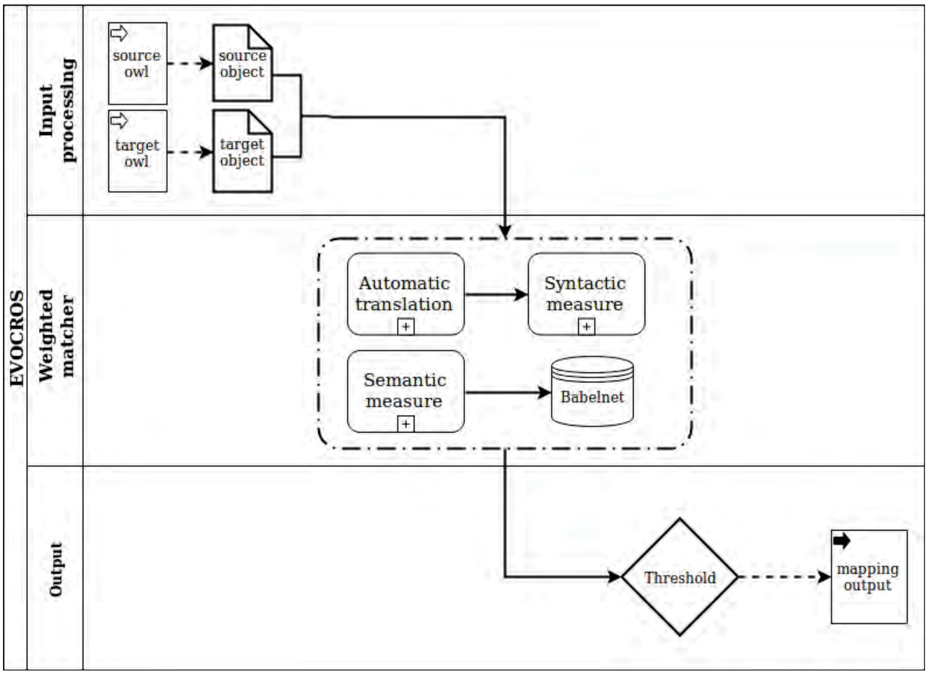


Fig. 1. EVOCROS workflow.

---

[3] Python 3 library to manipulate ontologies as objects.

Syntactic Similarity Measure. For syntactic similarity measure, the concept labels of both the source and target ontologies are first translated to a pivot language using automatic translation. We are using English as pivot language for OAEI 2018 though the tool accepts any language as pivot. The concepts are then compared by measuring the syntactic similarity via edit distance Levenshtein [3]) as a syntactic similarity measure.

Semantic Similarity Measure. Semantic similarity between terms is a metric to evaluate how similar two given terms are considering their meanings in a certain context. For example, the words "nail" and "hammer" are more similar considering the tool context than "nail" and "finger". On the other hand, when we consider the anatomy context, "nail" and "finger" are more similar than "nail" and "hammer".

For semantic similarity, we use the concept label in its original language, without any translation. There are a lot of algorithms to calculate semantic similarity. These algorithms usually explore an external resource such as vocabulary, dictionaries or thesauri to help computing the similarity between two words. EVOCROS explores a *Weighted Overlap* measure [6] relying on the neutral-domain semantic network *BabelNet* [5]. The tool retrieves from *Babelnet* the synsets of the concept labels of both source and target ontologies and compare them to measure the semantic similarity.

Our proposal generates cross-lingual ontology alignments taking into account the combination of semantic and syntactic similarity by computing the weighted average as follows:

Definition 1 (Composed Similarity). *Let sem(t_1, t_2) and sin(t_1, t_2) be the semantic similarity, and the syntactic one between the terms t_1 and t_2, respectively. We assume that the similarities are normalized between 0 and 1. Formally:*

$$simC(t_1, t_2) = \frac{\alpha sin(t_1, t_2) + \beta sem(t_1, t_2)}{\alpha + \beta} \tag{1}$$

where α and β are constants.

If the weighted similarity reaches a threshold, the concept pair is recorded to the output file, generated in RDF format. Otherwise, it is discarded.

1.3 Adaptations made for evaluation

EVOCROS uses a configuration file with the source and target ontologies, and their respective language. In order to participate in OAEI, we modified the tool to receive the source and target ontologies as input parameters and retrieve the ontology language from the *lang* XML tag. The bridge created for SEALS platform is written in Java and executed system calls to run the tool, written in Python 3. Although the tool executed locally using the SEALS client, there were issues during evaluation on SEALS platform and only local results are available in this report.

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

Alignment results are available at https://github.com/jmdestro/evocros-results.

2 Results

In this section, we describe the results obtained from local experiments using a sub-set of Multifarm with the same configuration used in OAEI 2018 evaluation.

2.1 Multifarm

Our experiments were based on ontologies from conference domain from the *MultiFarm dataset 2015* [4]. We used the reference mappings between the ontologies described in English and Spanish mapped into those concepts in the Portuguese Language.

Several weights for similarity measures and different similarity thresholds were evaluated locally. For OAEI 2018, only the following configuration was submitted: **threshold:** 0.66, **syntactic similarity weight:** 0.75, **semantic similarity weight:** 0.25. This was the configuration with the most interesting results. Table 1 presents the used configuration and the results for conference-conference alignment for languages spanish-portuguese (es-pt) and english-portuguese (en-pt).

Table 1. Cross-lingual mapping of conference-conference ontologies from MultiFarm.

Languages	Threshold	Syntactic similarity weight	Semantic similarity weight	Precision	Recall	F-measure
es-pt	0.66	0.75	0.25	0.68	0.33	0.44
en-pt	0.66	0.75	0.25	0.72	0.41	0.52

The choice of weights assigned to each similarity measure played an important role in the results. Tables 2 and 3 present the obtained results for different configurations. Considering the syntactical weights as 0.75 and 0.80 generated the best mappings, that is, they result in alignments with the greatest f-measure. Thus, our technique may be understood as a good alternative to syntactic or semantic only methods, and it might perform even better taking into account the correct parameters.

Table 2. MultiFarm alignment of Conference [ES] - Conference [PT] ontologies, using different threshold and weight.

Threshold	Syntactic weight	Semantic weight	Precision	Recall	F-measure
0.66	0.50	0.50	0.49	0.15	0.23
	0.33	0.67	0.40	0.10	0.16
	0.25	0.75	0.33	0.15	0.21
	0.20	0.80	0.30	0.15	0.20
	0.67	0.33	0.69	0.30	0.42
	0.75	0.25	0.68	0.33	0.44
	0.80	0.20	0.59	0.31	0.40
0.75	0.50	0.50	0.58	0.16	0.25
	0.33	0.67	0.48	0.16	0.24
	0.25	0.75	0.45	0.18	0.25
	0.20	0.80	0.40	0.17	0.24
	0.67	0.33	0.65	0.16	0.26
	0.75	0.25	0.75	0.31	0.44
	0.80	0.20	0.72	0.33	0.45
0.80	0.50	0.50	0.65	0.16	0.26
	0.33	0.67	0.58	0.16	0.25
	0.25	0.75	0.50	0.17	0.26
	0.20	0.80	0.45	0.18	0.25
	0.67	0.33	0.65	0.16	0.26
	0.75	0.25	0.65	0.16	0.26
	0.80	0.20	0.75	0.31	0.44
0.95	0.50	0.50	0.64	0.11	0.18
	0.33	0.67	0.67	0.15	0.24
	0.25	0.75	0.69	0.16	0.26
	0.20	0.80	0.65	0.16	0.26
	0.67	0.33	0.64	0.11	0.18
	0.75	0.25	0.64	0.11	0.18
	0.80	0.20	0.64	0.11	0.18

3 General comments

In this section, we discuss our results and the ways to improve the system.

3.1 Comments on the results (strength and weaknesses)

The tool had satisfactory results but the execution time was exceedingly long due to constant RestAPI calls to *Babelnet*. The results showed an influence of threshold: as the threshold rises, the precision also increases. It may be explained by considering equivalence of only those concepts with a high level of similarity. However, f-measure declines as the threshold increases because large values assigned to threshold make the algorithm disregards concepts that are equivalent, but somehow was assigned a lower level of similarity than expected by the threshold. As a result, the recall drops substantially, because many correct correspondences are ignored, and thus f-measure decreases. Empirically, we concluded

Table 3. MultiFarm alignment of Conference [EN] - Conference [PT] ontologies, using different threshold and weight.

Threshold	Syntactic weight	Semantic weight	Precision	Recall	F-measure
0.66	0.50	0.50	0.57	0.18	0.27
	0.33	0.67	0.42	0.21	0.28
	0.25	0.75	0.32	0.18	0.23
	0.20	0.80	0.28	0.17	0.21
	0.67	0.33	0.69	0.34	0.45
	0.75	0.25	0.72	0.41	0.52
	0.80	0.20	0.68	0.21	0.32
0.75	0.50	0.50	0.60	0.17	0.26
	0.33	0.67	0.52	0.23	0.32
	0.25	0.75	0.50	0.22	0.31
	0.20	0.80	0.43	0.21	0.28
	0.67	0.33	0.58	0.21	0.31
	0.75	0.25	0.70	0.15	0.25
	0.80	0.20	0.75	0.17	0.27
0.80	0.50	0.50	0.58	0.16	0.25
	0.33	0.67	0.57	0.23	0.32
	0.25	0.75	0.52	0.23	0.32
	0.20	0.80	0.50	0.22	0.31
	0.67	0.33	0.61	0.21	0.32
	0.75	0.25	0.61	0.09	0.15
	0.80	0.20	0.73	0.15	0.25
0.95	0.50	0.50	0.64	0.19	0.29
	0.33	0.67	0.61	0.21	0.32
	0.25	0.75	0.61	0.21	0.32
	0.20	0.80	0.61	0.21	0.32
	0.67	0.33	0.64	0.19	0.29
	0.75	0.25	0.64	0.07	0.13
	0.80	0.20	0.64	0.07	0.13

that the thresholds that generate the more accurate mappings were $\lambda = 0.66$ and $\lambda = 0.75$.

3.2 Discussions on the way to improve the proposed system

This was the first evaluation of the system and although there was issues during the evaluation phase of OAEI, preventing the system to be executed in SEALS platform, the local results are encouraging. Our main goals for future work are:

Reduce execution time: the tool has a long execution time due to constant RestAPI calls to *Babelnet* and needs to be optimized with local caches.

Bag of graphs: ontologies can be represented as graphs, thus allowing for partitioning [2] and comparison of sub-graphs. Bag-of-graphs [8] is a graph matching approach, similar to bag-of-words. It represents graphs as feature vectors, highly simplifying the computation of graph similarity and reducing execution time. We propose as future investigation to use a simple vector-based representation for graphs and investigate it for cross-lingual ontology matching.

3.3 Comments on OAEI

There were issues during the evaluation phase, preventing the system to participate in Multifarm track. For future editions of OAEI, we plan to participate submitting EVOCROS on the newly available HOBBIT platform, using a docker image, to ensure system compatibility during evaluation.

4 Conclusion

EVOCROS proposed an approach to cross-lingual ontology matching by combining semantic and syntactic similarity measures. This is the first participation of the system in OAEI. The evaluation with the Multifarm dataset confirmed the quality of mappings generated by our technique. For future work, we plan to improve our cross-lingual ontology alignment proposal considering different combinations of background knowledge, such as specific-domain thesauri to evaluate the semantic similarity. We also plan to further evaluate runtime optimization aspects to fix issues found during the evaluation phase.

Acknowledgements

This work has been supported by São Paulo Research Foundation (FAPESP): grants #2017/23522-3 and #2017/02325-5.

References

1. Destro, J.M., Dos Reis, J.C., Brito, A.M.C.R., Ricarte, I.L.M.: Influence of semantic similarity measures on ontology cross-language mappings. In: Proceedings of the Symposium on Applied Computing (SAC 2017). pp. 323–329. ACM (2017)
2. Hamdi, F., Safar, B., Reynaud, C., Zargayouna, H.: Alignment-based partitioning of large-scale ontologies. In: Advances in knowledge discovery and management, pp. 251–269. Springer (2010)
3. Levenshtein, V.I.: Binary codes capable of correcting deletions, insertions, and reversals. Soviet Physics Doklady **10**, 707710 (1966)
4. Meilicke, C., Garcia-Castro, R., Freitas, F., van Hage, W.R., Montiel-Ponsoda, E., de Azevedo, R.R., Stuckenschmidt, H., Sváb-Zamazal, O., Svátek, V., Tamilin, A., dos Santos, C.T., Wang, S.: Multifarm: A benchmark for multilingual ontology matching. Web Semantics: Science, Services and Agents on the World Wide Web **15**, 62–68 (2012)
5. Navigli, R., Ponzetto, S.P.: Babelnet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. Artificial Intelligence **193**, 217–250 (2012)

6. Pilehvar, M.T., Jurgens, D., Navigli, R.: Align, disambiguate and walk: A unified approach for measuring semantic similarity. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics. pp. 1341–1351 (2013)
7. Shvaiko, P., Euzenat, J.: Ontology matching: State of the art and future challenges. IEEE Transactions on Knowledge and Data Engineering **25**(1), 158–176 (Jan 2013). https://doi.org/10.1109/TKDE.2011.253
8. Silva, F.B., Tabbone, S., Torres, R.d.S.: Bog: A new approach for graph matching. In: Pattern Recognition (ICPR), 2014 22nd International Conference on. pp. 82–87. IEEE (2014)

FCAMapX results for OAEI 2018

Guowei Chen[1,2] and Songmao Zhang[2]

[1] University of Chinese Academy of Sciences
[2] Institute of Mathematics, Academy of Mathematics and Systems Science,
Chinese Academy of Sciences, Beijing, P.R. China
`gwch@amss.ac.cn`
`smzhang@math.ac.cn`

Abstract. FCAMapX is an automated ontology matching system based on Formal Concept Analysis, a mathematical model for analyzing individuals and structuring concepts. FCAMapX has succeeded in participating in three tracks of 2018 OAEI this year, including the Conference track, Anatomy, and Large Biomedical Ontologies. Based on our 2016 OAEI submission system FCA-Map which failed some large tasks within a designated time, we pursue improvements in efficiency and precision in FCAMapX. Concretely, we optimize the data structures for saving memory space and implement a more efficient algorithm for computing formal concept lattices. To favor precision, we tighten the condition for identifying lexical mappings and strengthen the structural validation to retrieve negative evidence for matches identified lexically and structurally. As a result, the running time for all the tasks has become less than an hour in our experimental setting; and in a majority of the cases, the precision and F-measure are both improved while the recall is lowered. Additionally, in comparison with other OAEI participants, FCAMapX has achieved the best or the second best F-measure and recall in most large biomedical ontology matching tasks.

1 Presentation of the system

Based on our 2016 OAEI participant system FCA-Map [1,3], this edition, called FCAMapX, pursues to improve the efficiency and precision.

1.1 State, purpose, general statement

In OAEI 2016, we submitted FCA-Map, a novel system based on Formal Concept Analysis to identify and validate mappings across ontologies, including one-to-one mappings and complex mappings. FCA-Map incrementally generates a total of three types of formal contexts and extracts mappings from the lattices derived. First, the token-based formal context describes how class names, labels, and synonyms share lexical tokens, leading to lexical mappings (anchors) across ontologies. Second, the relation-based formal context describes how classes are in taxonomic, partonomic and disjoint relationships with the anchors, leading to positive and negative structural evidence for validating the lexical matching. Third, the positive relation-based context can be used to discover structural mappings. The 2016 OAEI evaluation in the Anatomy, the Large Biomedical Ontologies, and the Disease and Phenotype track demonstrates the effectiveness of FCA-Map and its competitiveness with the top-ranked systems. For SNOMED-NCI(whole), the largest ontology matching task in OAEI, FCA-Map ranks first for recall and second for F-measure; ranks second for both F-measures of FMA-NCI and FMA-SNOMED, and obtains the best F-measures for most Disease and Phenotype tasks [3]. On the other hand, FCA-Map suffers from long running times due to the high complexity of deriving formal concept lattice in the Formal Concept Analysis formalism, which is a PSPACE-complete problem. Moreover, the performance of FCA-Map in terms of precision is relatively poorer than of recall and F-measure. We intend to address these two issues in the 2018 edition FCAMapX.

1.2 Specific techniques used

In order to improve the efficiency, we optimize the data structures for saving memory space and implement a more efficient algorithm Hermes [4] for computing formal concept lattice and Galois sub-hierarchy. The Hermes algorithm has an efficient running time of $O(min\{nm, n^{\alpha}\})$, where n is the number of objects or attributes, m the size of formal context, and n^{α} the time required to perform matrix multiplication (currently $\alpha = 2.376$). To improve the precision, we tighten the condition for identifying lexical mappings from the token-based lattice computed in the first step. Moreover, the second step for structural validation and the third step for structural mapping are swapped so that the positive and negative evidence can be retrieved for all mappings identified, lexically and structurally. This can favor precision as mappings with negative evidence are discarded.

1.3 Adaptations made for the evaluation

Similarly to our previous edition, our SEALS submission included precomputed word variants originated from UMLS[5] for mapping biomedical ontologies. Moreover, in order to augment the performance of FCAMapX in mapping ontologies in general purpose domains like those of the Conference track, we used the synsets of WordNet[6] in the first step for identifying synonymous terms. Property names in the Conference ontologies are also taken into account when constructing the token-based formal context for lexical mapping.

1.4 Link to the system and parameters file

SEALS wrapped version of FCAMapX for OAEI 2018 is available at `https://drive.google.com/open?id=1-0upxrcPbu5OVJAJn-DtTOUMOh3QDriM`.

1.5 Link to the set of provided alignments

The results obtained by FCAMapX for OAEI 2018 are available at `https://drive.google.com/open?id=1DzRD_90O3YwoGpW5FJL9vSy_f1Ia0YZo`

2 Results

In this section, we present our evaluation results obtained by running FCAMapX over the tracks of *Anatomy*, *Conference*, and *Large Biomedical Ontologies*. Tests were performed using a desktop computer with 16 GB of RAM and Intel® Core$^{\text{TM}}$ i7-8700 CPU @ 3.20GHz.

2.1 The OAEI 2018 Anatomy Track

The anatomy track consists of the Adult Mouse Anatomy (2744 classes) and a fragment of the NCI Thesaurus (3304 classes) for describing the human anatomy. Compared with our 2016 version, FCAMapX has improved the precision from 0.932 to 0.941, whereas the recall is decreased from 0.837 to 0.791, leading to a drop of the F-Measure from 0.882 to 0.860, as shown in Table 1).

Table 1. Results for Anatomy track

Task	Precision	Recall	F-Measure	Runtime (s)
MA-NCI	0.941	0.791	0.860	11.811

2.2 The OAEI 2018 Conference Track

The Conference 2018 Track contains 16 ontologies describing the domain of conference organizations. These ontologies are of smaller scale with limited classes and semantic relations, for which our approach can be ineffective, as analyzed in [3]. In this edition, we add external knowledge source WordNet and the results are listed in Table 2. Taking advantage of the additional synonyms defined in WordNet for general purpose domains, FCAMapX has increased the average recall from 0.52 to 0.582 and the average F-measure from 0.61 to 0.62, while the precision drops from 0.75 to 0.698.

2.3 The OAEI 2018 Large Biomedical Ontologies 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 of both large-scale and semantic richness. The results obtained by FCAMapX are depicted in Table 3. Except for FMA-NCI (small), in all other five tasks, FCAMapX has managed to increase the precision as well as

Table 2. Results for Conference track

Task	Precision	Recall	F-Measure	Runtime (s)
cmt-conference	0.563	0.600	0.581	1.194
cmt-confOf	0.667	0.375	0.480	0.291
cmt-edas	0.615	0.615	0.615	0.391
cmt-ekaw	0.556	0.455	0.500	0.254
cmt-iasted	0.500	1.000	0.667	0.546
cmt-sigkdd	0.750	0.750	0.750	0.222
conference-confOf	0.818	0.600	0.692	0.243
confenrece-edas	0.600	0.529	0.562	0.355
conference-ekaw	0.619	0.520	0.565	0.273
conference-iasted	0.364	0.286	0.320	0.466
conference-sigkdd	0.750	0.600	0.667	0.223
confOf-edas	0.846	0.579	0.687	0.304
confOf-ekaw	0.857	0.600	0.706	0.24
confOf-iasted	0.857	0.667	0.750	0.403
conOf-sigkdd	1.000	0.571	0.727	0.193
edas-ekaw	0.647	0.478	0.550	0.343
edas-iasted	0.727	0.421	0.533	0.48
edas-sigkdd	0.875	0.467	0.609	0.293
ekaw-iasted	0.462	0.600	0.522	0.467
ekaw-sigkdd	0.778	0.636	0.700	0.235
iasted-sigkdd	0.813	0.867	0.839	0.534

the F-measure while the recall values are lowered. Take FMA-SNOMED (whole) for example, the precision is 1.8 times of the 2016 version and the F-measure 1.4 times. More importantly, in our own experimental setting, FCAMapX finished all tasks in the Large Biomedical track within 2 hours as required by 2016 OAEI, whereas our 2016 system failed the three Whole tasks. For the largest task SNOMED-NCI (whole), our previous version ran about 13 hours as reported in [3], and by FCAMapX, the time has been downsized to 0.95 hours.

Table 3. Results for Large Biomedical track

Task	Precision	Recall	F-Measure	Runtime (s)
FMA-NCI (small)	0.948	0.911	0.929	73.692
FMA-NCI (whole)	0.665	0.841	0.743	1171.62
FMA-SNOMED (small)	0.955	0.815	0.879	125.791
FMA-SNOMED (whole)	0.819	0.762	0.789	2179.924
SNOMED-NCI (small)	0.878	0.703	0.781	1039.138
SNOMED-NCI (whole)	0.796	0.680	0.733	3418.672

As reported by OAEI [3], out of the six tasks in the track, FCAMapX ranks first for three and second for two tasks in terms of recall; and for F-measure, FCAMapX ranks first for two and second for three tasks.

3 General comments

This is the second time that we participate in the OAEI campaign with our Formal Concept Analysis based systems. The main goal is to improve the efficiency in regard to our 2016 edition which failed to finish within the designated time for three tasks in Large Biomedical Ontologies track. This has been accomplished by FCAMapX. At the same time, strengthening the structural validation of mappings has yielded higher precisions which can lead to better F-measure values.

3.1 Comments on the results

FCAMapX has succeeded in participating in three tracks this year, including the Conference track, Anatomy, and Large Biomedical Ontologies. The running time for all the tasks has become less than an hour now in our experimental setting. In a majority of the cases, the precision and F-measure are both improved while the recall is lowered. That FCAMapX performs unsatisfactorily for FMA-NCI (small) in comparison with our 2016 system deserves a further explanation.

3.2 Discussions on the way to improve the proposed system

We intended to run FCAMapX on the Disease and Phenotype track where our previous 2016 system performs competitively [1,3]. The results in our own setting against the consensus alignments with vote 3 are listed in Table 4, where the matching tasks involve the Human Phenotype (HP) Ontology, the Mammalian Phenotype (MP) Ontology, the Human Disease Ontology (DOID), and the Orphanet and Rare Diseases Ontology (ORDO). Note that these results cannot be compared with our 2016 system, as the version and source of the four ontologies are different from the ones used in 2016 [4].

Unfortunately, FCAMapX failed this track with errors as reported by the OAEI evaluation. This indicates that the quality of the system shall be improved.

Table 4. Results for Disease and Phenotype track

Task	Precision	Recall	F-Measure	Runtime (s)
HP-MP	0.848	0.760	0.802	2368.376
DOID-ORDO	0.869	0.729	0.793	450.134

---

[3] http://www.cs.ox.ac.uk/isg/projects/SEALS/oaei/2018/results/

[4] http://oaei.ontologymatching.org/2018/phenotype/

3.3 Comments on the OAEI procedure

With our participating experience this year, we find that OAEI is well organized in an efficient way and organizers helpful. Various tracks have different levels of difficulty, which is challenging and appealing, and the SEALS platform is very convenient to use.

4 Conclusions

In this paper, we present FCAMapX as an improved version of our 2016 OAEI system FCA-Map. The improvement mainly lies in the efficiency, as illustrated by the dramatic drop of running times, for instance from 13 to 1 hour for the largest OAEI task. The second improvement is on the mapping precision which normally causes the F-measure to rise. Compared with other OAEI participants, FCAMapX has achieved the best or the second best F-measure and recall in five out of the six large biomedical ontology matching tasks. Despite these, our system still has a long way to go in terms of covering all OAEI tracks, especially those instance matching tasks for which the Formal Concept Analysis formalism has a potential to prevail with its capability of clustering commonalities among individuals.

Acknowledgements

This work has been supported by the National Key Research and Development Program of China under grant 2016YFB1000902, and the Natural Science Foundation of China grant 61621003.

References

1. Zhao, M., & Zhang, S. (2016, December). FCA-Map results for OAEI 2016. In OM@ ISWC (pp. 172-177).
2. Zhao, M., & Zhang, S. (2016, December). Identifying and validating ontology mappings by formal concept analysis. In OM@ ISWC (pp. 61-72).
3. Zhao, M., Zhang, S., Li, W., & Chen, G. (2018). Matching biomedical ontologies based on formal concept analysis. Journal of biomedical semantics, **9**(1), 11.
4. Berry, A., Huchard, M., Napoli, A., & Sigayret, A. (2012, October). Hermes: an efficient algorithm for building Galois sub-hierarchies. In CLA: Concept Lattices and their Applications (pp. 21-32). Universidad de Malaga.
5. Lindberg DA, Humphreys BL, McCray AT, et al. The unified medical language system. IMIA Yearbook. 1993;32(4):28191.
6. G. A. Miller. WordNet: A Lexical Database for English. Communications of the ACM, 38(11):3941, 1995.
7. de Souza, K.X.S., Davis, J.: Aligning ontologies and evaluating concept similarities. In: OTM Confederated International Conferences On the Move to Meaningful Internet Systems, Springer (2004) 10121029

8. Guan-yu, L., Shu-peng, L., et al.: Formal concept analysis based ontology merging method. In: Computer Science and Information Technology (ICCSIT), 2010 3rd IEEE International Conference on. Volume 8., IEEE (2010) 279282
9. Obitko, M., Snsel, V., Smid, J.: Ontology design with formal concept analysis. CLA 128(3) (2004) 13771390
10. Stumme, G., Maedche, A.: Fca-merge: Bottom-up merging of ontologies. In: IJCAI. Volume 1. (2001) 225230
11. Wille, R.: Restructuring lattice theory: an approach based on hierarchies of concepts. In: Ordered sets. Springer (1982) 445470
12. Xu, X., Wu, Y., Chen, J.: Fuzzy fca based ontology mapping. In: 2010 First International Conference on Networking and Distributed Computing, IEEE (2010) 181185
13. Zhang, S., Bodenreider, O.: Experience in aligning anatomical ontologies. International journal on Semantic Web and information systems 3(2) (2007) 1

Holontology : results of the 2018 OAEI evaluation campaign

Philippe Roussille, Imen Megdiche, Olivier Teste, Cassia Trojahn

Institut de Recherche en Informatique de Toulouse, Toulouse, France
{firstname.name}@irit.fr

Abstract. This paper presents the results obtained by the Holontology (Holistic ontology matcher) system in the OAEI 2018 evaluation campaign. We describe here the results in the Anatomy, Conference, Taxon and Knowledge Graph tracks. We report a general discussion on the results and on the future improvements of the system.

1 Presentation of the system

1.1 State, purpose, general statement

Holontology is a modular system based on the LPHOM system (Linear Program for Holistic Ontology Matching) [1]. As its predecessor LPHOM, the system remains a holistic ontology matching system i.e., matching multiple ontologies simultaneously. Although the system has been designed to deal with holistic matching, it is able as well to deal with pairwise ontology matching, as we consider this task as a particular case of the holistic one, as described here.

The system treats the ontology matching problem, at schema-level, as a combinatorial optimization problem. The problem is modelled through a linear program extending the maximum-weighted graph matching problem with linear constraints (matching cardinality, structural, and coherence constraints).

1.2 Specific techniques used

The way the system works is detailed in the following steps:

1. The first step of Holontology is to load the ontologies and translate them into an internal structure, which can be annotated and edited, so that later treatment is done in an efficient way. Each ontology is loaded independently, and its hierarchical structure is conserved in a format based on the three possible basic blocks of ontologies: classes and properties (object and data). Each of them is represented as a *Node*, and depending on their subtype, they are subclassed into a *ClassNode*, *ObjectPropertyNode* or *DataPropertyNode*. These nodes are then enriched with *AnnotationData*.
2. The second step consists in a pre-processing step. Here, we expand camel case and title case into proper names. As the loading of the ontologies is done only once, we take advantage of Java ways of storing references, so we have no need to translate and cut matrices. We can use our hierarchies both for storage and computation.

3. The third step computes a cartesian product between all the entities of same type (classes and properties) in order to build a similar linear program using the same framework as LPHOM, as described below.
4. The fourth step leads a second wave of pre-processing, in such a way that new relations are created. For example, this is the step where we compute tense similarity (seeing whether or not an object property is passive or active towards its classes). More precisely, for each property, we compute its polarity using tense analysis, e.g. *Author reviews Paper* (active) vs. *Paper reviewed by Author* (passive). This allows *reviews* matching *reviewed by reversed*.
5. The fifth step applies a combination of different similarity measures (exact match, Levenstein, Jaccard, and Lin), drawing both from the *AnnotationData* build in the previous steps, and the constraints based on ontologies themselves.We have tested the combination of similarity measures and the results reported here are in terms of ‘exact match’ over the pre-processed entity naming and annotations (considering the exact match between tokens). As expected, we obtain good values in terms of precision, as better discussed below.
6. The sixth step converts the given constraints in a form that can be used by a solver. We choose CPLEX for its ability to perform optimizations, and we manage probable thresholds if needed. Unlike LPHOM, we provide all the constraints, and we do not apply any cutting in this step.
7. In the seventh step, we take the given result from the solver and convert it in an alignment that can be exported in RDF.
8. In the (optional) eight step, we combine the alignment result and run step five to seven if needed to reinforce the obtained results.

1.3 Adaptations made for the evaluation

Due to a huge incompatibility between some libraries used in the SEALS client and ours, we had to create a fully executable jar in the conf/ directory, and call this jar from the SEALS bridge. The bridge grabbed the URI of the two ontologies passed as arguments, and wrote them to a file called “bridge-ontologies.url”, then called the holontology.jar in the conf directory, waiting for its full execution. holontology.jar opens the created “ontologies.url”, reads the two URIs, and does the processing. The result is written in the “result.alignment” file. The bridge then reads the result file and returned it to the SEALS client.

1.4 Link to the system and parameters file

Holontology, as its predecessor LPHOM, is not available as an open-source. The version we present here is meant to be more modulable than LPHOM, and handles the problem differently, by insisting on annotating data and using structures instead of matrices. The .zip for the SEALS jar can be retrieved at `https://cloud.irit.fr/index.php/s/gReZo8yaRDqdmjk`

1.5 Link to the set of provided alignments

The generated alignments are available at `https://cloud.irit.fr/index.php/s/hv3oALXN6fHuZWi`.

2 Results

The reader can refer to the OAEI web pages for the results of Holontology in the Anatomy, Conference, Taxon and Knowledge Graph tracks as well a comparison with other participants. Here, we provide a first discussion and comments on our results.

2.1 Anatomy track

Our results for the Anatomy track are summarised in Table 2.1. Compared to the evaluation of LPHOM in OAEI 2016 [1], we observe that globally the quality of results decreases, for instance the F-measure looses 0.3 points. These results can be explained by two choices in Holontology. First we only use exact match (we observe that Holontology returns only 456 alignments compared to LPHOM which returns 1555 alignments). Second we do not cut results according to that (i.e, we do not apply any threshold). However, we observe that Holontology is 8 times faster that LPHOM.

Matcher	Runtime	Size	Precision	F-Measure	Recall	Recall+	Coherent
Holontology	265	456	0.976	0.451	0.294	0.005	-

Table 1. Results for Anatomy track.

2.2 Conference track

Our results for the Conference track are summarised in Table 2.2. Contrarily to the anatomy track, the results of Holontology are better than the results of LPHOM for the conference track. Globally, Holontology gains in recall for the different tasks in this track. We can explain that by the different pre-processing strategies that have been implemented in Holontology compared to LPHOM. By comparing the different tasks, the tool needs additional efforts to handle data and object properties that occur in the M2 tasks.

2.3 Complex track (Taxon task)

Hontology is not able to deal with complex matching and has not been initially registered to this track. However, given that only 3 systems have been registered to the complex track, the organisers have also run the systems registered to Anatomy and Conference on the complex datasets. Hence, the results reported for Hontology are in terms of simple generated alignments.

---

[1] http://oaei.ontologymatching.org/2016/results/anatomy/index.html

Track	Rank	Prec.	F.5-measure	F1-measure	F2-measure	Recall
ra1-M1	8/13	0.88	0.78	0.67	0.59	0.54
ra1-M2	11/13	0.22	0.17	0.13	0.1	0.09
ra1-M3	10/13	0.78	0.69	0.59	0.52	0.48
ra2-M1	6/13	0.81	0.72	0.62	0.54	0.5
ra2-M2	11/13	0.07	0.05	0.03	0.02	0.09
ra2-M3	7/13	0.74	0.65	0.55	0.48	0.44
rar2-M1	9/13	0.8	0.72	0.63	0.56	0.52
rar2-M2	11/13	0.22	0.17	0.13	0.1	0.09
rar2-M3	9/13	0.73	0.65	0.56	0.49	0.45

Table 2. Results for the Conference track.

Our results for the Taxon task are summarised in Table 2.3. We have been obtained intermediate results, with a precision up to 0.22. However, for the set of given queries to be translated with the help of the generated alignments, our alignments were not useful, QWR (Query Well Rewritten) measure of 0.

Time (s)	output corres.	eval. corres.	correct corres.	Global Prec.	Average Prec.	(1:1)	(1:n)	(m:n)	QWR
965	44	13	3	0.23	0.22	44	0	0	0.00

Table 3. Results for taxon task in the Complex track.

2.4 Knowledge graph track

Our results for the Knowledge graph track are summarized in Tables 2.4 and 2.4. For this track, Holontology proceeded faster than the other systems (including the baseline). However, it has not be able to deal with properties, probably, as the track organisers explained, because all properties are typed as *rdf:Property* and not subdivided into *owl:DatatypeProperty* and *owl:ObjectProperty*).

		class				**overall**			
Time	**#tracks**	**Size**	**Prec.**	**F-m.**	**Rec.**	**Size**	**Prec.**	**F-m.**	**Rec.**
318	9	16.8	0.80 (0.80)	0.83 (0.83)	0.87 (0.87)	18.8	0.80 (0.80)	0.17 (0.17)	0.10 (0.10)

Table 4. Global results for Knowledge graph track.

3 General comments

Despite the fact that Hontology is an extended version of LPHOM that has participated in OAEI 2016, for its first participation Hontology has relative intermediate results. Table 3 summarises the performance of Hontology in terms of ranking of the best systems in each task.

Track	Time	Size	Prec.	F-m.	Rec.
darkscape∼oldschoolrunescape	23	42	0.75	0.22	0.13
runescape∼darkscape	38	62	0.93	0.25	0.14
runescape∼oldschoolrunescape	35	47	0.62	0.22	0.13
heykidscomics∼dc	67	4	1.00	0.10	0.05
marvel∼dc	26	4	1.00	0.21	0.12
marvel∼heykidscomics	61	4	1.00	0.11	0.06
memory-alpha∼memory-beta	26	2	0.00	0.00	0.00
memory-alpha∼stexpanded	21	2	0.00	0.00	0.00
memory-beta∼stexpanded	21	2	0.00	0.00	0.00

Table 5. Track results for Knowledge graph track.

Track	Rank
Anatomy	14/14
Taxon	3/7
Knowledge graph	6/8 (overall)
Knowledge graph	1/8 (classes)
Conference	9.1/13 (average)

Table 6. Global rankings for Holontology per track.

With respect to LPHOM, Hontology is a modular system that optimises the ontology structures in memory. As for LPHOM, we model and express the matching problem through a set of constraints (cardinality, structural, and coherence constraints) applied on the results of a pre-processing and exact matching steps. We do not have applied any threshold on the generated alignments. As expected, using an exact match on pre-processed entity naming and comments may improve precision in detriment of recall. With respect to these aspects, we plan to improve the criteria of selection of similarity measures and thresholds for our future participation, in particular with the aim of improving recall.

Other points include the fact that Hontology is a system designed to deal with holistic ontology matching at schema-level. Hence, it was not able to generated alignments for the tasks involving instance matching. We plan to implement instance matching strategies in future versions of the system. Furthermore, despite our optimisation over LPHOM, our system was not able to deal at all with the large ontologies in the LargeBio and Phenotype tasks. We note, however, that we could deal with the Complex Taxon task in terms of volume because we do not treat the instances. We plan to address these points in the future.

With respect to the OAEI procedure, we focus on the tracks based on SEALS. However, as stated above, we have encountered problems for dealing with the incompatibilities of package versions in the SEALS dependencies. We have implemented a non-ideal solution and hope for the next evaluation this kind of issue will be fixed.

Finally, Hontology has been initially designed to deal with hoslistic ontology matching. However, there is no track in the campaign proposing the evaluation of such kind of matching approaches. In the future, it could be interesting to have a dedicated holistic track.

4 Conclusions

This paper has introduced the Hontology system and discussed the main points on the results of its first participation in the OAEI campaigns. We have as well pointed out some directions for future improvements.

Acknowledgements

Cassia Trojahn has been partially supported by the French CIMI Labex project IBLiD (Integration of Big and Linked Data for On-Line Analytics).

References

1. I. Megdiche, O. Teste, and C. Trojahn. An extensible linear approach for holistic ontology matching. In *Proceedings of the 15th International Semantic Web Conference (ISWC 2016), Kobe, Japan, October 18, 2016.*, pages 393–410. Springer International Publishing, 2016.

KEPLER at OAEI 2018

Marouen KACHROUDI[1], Gayo DIALLO[2], and Sadok BEN YAHIA[1]

[1] Université de Tunis El Manar, Faculté des Sciences de Tunis
Informatique Programmation Algorithmique et Heuristique
LR11ES14, 2092, Tunis, Tunisie
`{marouen.kachroudi,sadok.benyahia}@fst.rnu.tn`
[2] BPH Center - INSERM U1219, Team ERIAS & LaBRI UMR5800,
Univ. Bordeaux, F-33000, France
`gayo.diallo@u-bordeaux.fr`

Abstract. This paper presents and discusses the results of the KEPLER system for the 2018 edition of the Ontology Alignment Evaluation Initiative (OAEI 2018). The implemented approach is based on the exploitation of three different strategies including Information Retrieval (IR) inspired algorithm for terminological based alignment computation. For scaling up, KEPLER implements a partitioning approach, while for the management of multilingualism, KEPLER develops a well-defined strategy based on the use of a translator and structural alignment computation. This is the second year of participation and the results are encouraging.

1 Presentation of the system

A substantial growth of the semantic Web users create and update knowledge resources all over the world using various conceptualizations. These knowledge resources are used for annotating available online data. This process is nowadays being accelerated due to few initiatives which encourage to make data available in a comprehensive way for agents [1]. However, as they are annotated by different conceptual schemes, an effort is needed to make them interoperable. As of a solution, ontology alignment process is applied in order to identify bridges between the heterogeneous knowledge resources (ontologies, structured vocabularies, etc.) which play the role of semantic background for the available data. This process facilitates the share and reuse of these resources [2].

KEPLER is an ontology alignment system which deals with the key challenges related to heterogeneous ontologies on the semantic Web. It is grounded from previous approaches [3–6] and relies on several alignment strategies summarized in the following sections. It is designed to discover alignments for both common size and large scale ontologies as well as computing alignments in a multilingual context.

1.1 State, purpose, general statement

KEPLER exploits, besides classic techniques [7], an external resource, *i.e.*, a translator in order to deal with multilingualism.

1.2 Specific techniques used

The main idea of KEPLER is to exploit the expressiveness of the OWL language to detect and compute the similarity between entities of two given ontologies through six complementary modules as presented in Figure 1.

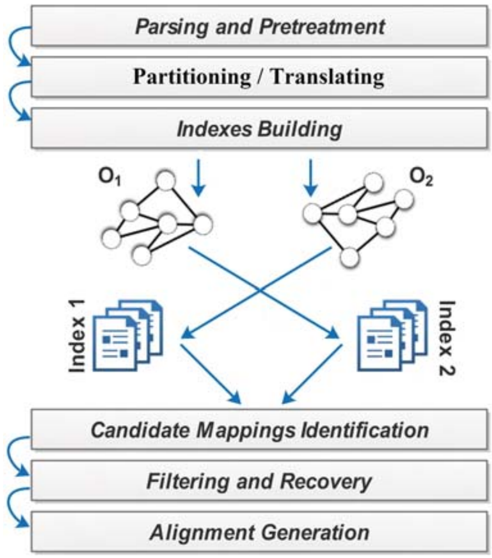


Fig. 1. KEPLER workflow.

Entities are described using OWL primitives with their semantics. An ontology is seen as a semantic graph where entities are nodes connected by links (the predicates). These links have specified semantics. The alignment workflow is detailed as follows.

Parsing and pretreatment: this module extracts the ontological entities initially represented by a primary form of lists. In other words, at the parsing stage, the main goal is to transform an OWL ontology in a well defined structure that preserves and highlight all the information contained in processed ontology. It has a significant impact on the results of the similarity computation thereafter. The result is a set of entities names and their associated descriptions.

Partitioning: KEPLER follows a divide and conquer strategy. Therefore, this module aims at splitting ontologies into smaller parts to support the alignment task [8]. Consequently, partitioning a set $\mathcal{B}(\mathcal{C})$ is to find subsets $\mathcal{B}_1$, $\mathcal{B}_2$,..., $\mathcal{B}_n$, encompassing semantically close elements bound by a relevant set of relationships, *i.e.*, $\mathcal{O} = \bigcup\{\mathcal{B}_1, \mathcal{B}_2, ..., \mathcal{B}_n\}$, where $\mathcal{B}_i$ is an ontological block, and n is the resulting number of extracted blocks. Hence, we can define an ontological portion as a reduced ontology that could be extracted from another larger one by splitting up the latter according to its constituents : structures and semantics. One way to obtain such a partitioning is to maximize the relationships inside a block while minimizing the relationship between the blocks themselves. The resulting partitioning quality can be evaluated using different criteria:

- *The size of the generated blocks*: that must have a reasonable size, *i.e.*, a number of elements that could be handled by an alignment tool;
- *The number of the generated blocks*: this number should be as small as possible to limit the number of block pairs to be aligned latter;
- *The compactness degree of a block*: a block is said to be substantially compact if relations (lexical and structural ones) are stronger inside the block and lower outside it.

Translation : in order to deal with multilingualism, two alternatives are followed: *i*) either considering one of the languages of the input ontologies as a pivot, therefore translating the second one to this chosen pivot; *ii*) choosing a pivot language and translate the inputs ontologies to this pivot. Further to these alternatives, an external resource, *i.e.*, WordNet[3] is used. Therefore the pivot language used by KEPLER is the English language. The translation process is performed usinig the Microsoft Bing [4] translator.

Indexing : one of the issue in Ontology Alignment is the cost of computing the similarity between all the entities of the input ontologies. To deal with this issue, the indexing strategy is one of the novelties of our approach. It consists in reducing the search space through the use of techniques borrowed from the IR domain. An effective search strategy is implemented on top of the built indexes of the two input ontologies. To enable faster searching, the driving idea that was previously used in the ServOMap system [9] is to perform the analysis of the ontologies in advance and store it in an optimized format for the search.

Candidate Mappings Identification : the role of this module is to find the entities in common between the indexes. Once the indexes are set up, the querying step is activated. To do so, the querying strategy implemented satisfies both the terminology search and semantic aspects at once. Indeed, the task is querying documents in a vector space that contains a set of ontological entities and their synonyms obtained via WordNet for each Ontology. It is worthy to mention that indexes querying is done in both senses (each ontology plays successively the role of querying component).

Filtering and Recovery: the filtering module consists of two complementary submodules, each one is responsible of a specific task in order to refine the set of primarily identified candidates mappings. At this stage, once the list of candidates is ready, the alignment method uses a first filter. This filter eliminates the redundancy between these candidates by eliminating possible duplicates. In addition, there is always the concern about *false positives*. The second filter eliminates *false positives* candidates. This filter is applied to what is called *partially* redundant entities. An entity is considered as *partially* redundant if it belongs to two different mappings. Being given three ontological

---

[3] https://wordnet.princeton.edu/

[4] https://www.bing.com/translator

entities e_1, e_2 and e_3, if on the one hand, e_1 is aligned to e_2, and secondly, e_1 is aligned to e_3, this last alignment is qualified as doubtful. As the KEPLER system generates $(1 : 1)$ mappings, an entity cannot belongs to several mappings. Therefore, given the topology of two suspicious entities (e_3 neighbors with e_1 neighbors, e_2 neighbors with e_1 neighbors) with respect to the redundant entity e_1, the idea is to retain the couple having the highest topological proximity value. All candidates are subject to this filter before to generate the final alignment.

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

2 Results

In this section, we present the results obtained by KEPLER system for the OAEI 2018 edition.

2.1 Anatomy track

This track consists in two real world ontologies to be matched, the source ontology describing the Adult Mouse Anatomy (with 2744 classes) and the target ontology is the NCI Thesaurus describing the Human Anatomy (with 3304 classes). For this track, KEPLER succeeded to extract 74% of correct mappings with a precision of 95% and recall of 74%. KEPLER handled easyly the input ontologies of this track thanks to the partionning module $Ontopart$ [10, 8]

2.2 Conference track

The conference track consists of 15 ontologies from the conference organization domain and each ontology must be matched against every other ontologies. The dataset describes the domain of organizing conferences from different perspectives. Precision values for to evaluation settings are respectively 76% and 58%. Recall values are 48% and 68%.

2.3 Multifarm track

The Multifram dataset is composed of a subset of the Conference track, translated in nine different languages (*i.e.*, Chinese, Czech, Dutch, French, German, Portuguese, Russian, Spanish and Arabic). With a special focus on multilingualism, it is possible to evaluate and compare the performance of alignment approaches through these test cases. Based on several previous contributions [11–16], the designed main goal of the MultiFarm track is to evaluate the ability of the alignment systems to deal with multilingual ontologies. It serves the purpose of evaluating the strength and weakness of a given system across languages. In the *different ontologies* setting, KEPLER is ranked second with a recall value of 0.21 and a precision value of 0.40. Whereas in the *same ontologies* setting, it lasted at the first place with a recall value of 0.36, and a precision value of 0.85.

2.4 Complex track

KEPLER succeeds in the best case, to obtain 27% of recall and a precision of 100%.

2.5 Large Biomedical Ontologies track

In the scalability register, this track consists in 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. The Large BioMed Track consists of three matching problems, *i.e.*, (1) FMA-NCI matching problem, (2) FMA-SNOMED matching problem and (3) SNOMED-NCI matching problem. KEPLER succeeded providing results for the (*Task 1: FMA-NCI small fragments*)`[Precision : 0.96 / Recall : 0.83]` and task 3 of the track (*FMA-SNOMED small fragments*) with a Precision of 0.82 and Recall of 0.42.

2.6 Phenotype

In the Phenotype track, the system succeeded in processing only the DOID-ORDO subcase by identifying 1824 matches for 1237 expected ones, `[Precision : 0.86 / Recall : 0.59]`.

3 Conclusion

In this paper, we briefly described the alignment system KEPLER with comments of the results obtained according to the OAEI 2018 tracks, corresponding to the SEALS platform evaluation modality. Several observations regarding these results were highlighted, in particular the impact of the elimination of any ontological resource on the similarity values. KEPLER is an ongoing work which borrows its idea from two previous systems, CLONA [15] and SERVOMAP [9]. It showed promising results for this second participation. As future work, the idea is to support the instance based ontology alignment in a wider range and contexts [17]. We have dealt with this issue before [18, 19], but the test base update imposes other challenges in terms of the used ontological languages and the evolutive semantic description formalisms.

References

1. Berners-Lee, T.: Designing the web for an open society. In: Proceedings of the 20^{th} International Conference on World Wide Web (WWW2011), Hyderabad, India (2011) 3–4
2. Suchanek, F.M., Varde, A.S., Nayak, R., Senellart, P.: The hidden web, xml and semantic web: A scientific data management perspective. Computing Research Repository (2011) 534–537
3. Kachroudi, M., Ben Moussa, E., Zghal, S., Ben Yahia, S.: Ldoa results for oaei 2011. In: Proceedings of the 6th International Workshop on Ontology Matching (OM-2011) Colocated with the 10^{th} International Semantic Web Conference (ISWC-2011), Bonn, Germany (2011) 148–155

4. Kachroudi, M., Zghal, S., Ben Yahia, S.: Alignement d'ontologies en utilisant des ressources externes : Linked data. In: Actes des 4$^{\text{èmes}}$ Journées Francophones sur les Ontologies (JFO'2011), Montréal, Canada (2011) 259–264
5. Diallo, G.: Efficient building of local repository of distributed ontologies. In: Proceedings of the Seventh International Conference on Signal-Image Technology and Internet-Based Systems, SITIS 2011, Dijon, France, November 28 - December 1, 2011. (2011) 159–166
6. Dramé, K., Diallo, G., Delva, F., Dartigues, J., Mouillet, E., Salamon, R., Mougin, F.: Reuse of termino-ontological resources and text corpora for building a multilingual domain ontology: An application to alzheimer's disease. Journal of Biomedical Informatics **48** (2014) 171–182
7. Euzenat, J., Mocan, A., Scharffe, F.: Ontology alignments: an ontology management perspective. Ontology management: semantic web, semantic web services, and business applications (2008)
8. Kachroudi, M., Zghal, S., Ben Yahia, S.: Ontopart: at the cross-roads of ontology partitioning and scalable ontology alignment systems. International Journal of Metadata, Semantics and Ontologies **8**(3) (2013) 215–225
9. Diallo, G.: An effective method of large scale ontology matching. Journal of Biomedical Semantics **5(44) doi:10.1186/2041-1480-5-44** (2014)
10. Kachroudi, M., Hassen, W., Zghal, S., Ben Yahia, S.: Large ontologies partitioning for alignment techniques scaling. In: Proceedings of the 9^{th} International Conference on Web Information Systems and Technologies (WEBIST), 8-10 May, Aachen, Germany (2013) 165–168
11. Kachroudi, M., Ben Yahia, S., Zghal, S.: Damo - direct alignment for multilingual ontologies. In: Proceedings of the 3^{rd} International Conference on Knowledge Engineering and Ontology Development (KEOD), 26-29 October, Paris, France (2011) 110–117
12. Kachroudi, M., Zghal, S., Ben Yahia, S.: When external linguistic resource supports cross-lingual ontology alignment. In: Proceedings of the 5^{th} International Conference on Web and Information Technologies (ICWIT 2013), 9-12, May, Hammamet, Tunisia (2013) 327–336
13. Kachroudi, M., Zghal, S., Ben Yahia, S.: Using linguistic resource for cross-lingual ontology alignment. International Journal of Recent Contributions from Engineering **1**(1) (2013) 21–27
14. Kachroudi, M., Zghal, S., Ben Yahia, S.: Bridging the multilingualism gap in ontology alignment. International Journal of Metadata, Semantics and Ontologies **9**(3) (2014) 252–262
15. El Abdi, M., Souid, H., Kachroudi, M., Ben Yahia, S.: Clona results for oaei 2015. In: Proceedings of the 12^{th} International Workshop on Ontology Matching (OM-2015) Colocated with the 14^{th} International Semantic Web Conference (ISWC-2015). Volume 1545 of CEUR-WS., Bethlehem (PA US) (2015) 124–129
16. Kachroudi, M., Diallo, G., Ben Yahia, S.: Initiating cross-lingual ontology alignment with information retrieval techniques. In: Actes de la 6$^{\text{ème}}$ Edition des Journées sur les Ontologies (JFO'2016), Bordeaux, France (2016) 57–68
17. Kachroudi, M., Diallo, G., Ben Yahia, S.: On the composition of large biomedical ontologies alignment. In: Proceedings of the 7^{th} International Conference on Web Intelligence, Mining and Semantics, WIMS 2017. (2017) 1–10
18. Damak, S., Souid, H., Kachroudi, M., Zghal, S.: Exona results for oaei 2015. In: Proceedings of the 12^{th} International Workshop on Ontology Matching (OM-2015) Colocated with the 14^{th} International Semantic Web Conference (ISWC-2015). Volume 1545 of CEUR-WS., Bethlehem (PA US) (2015) 145–149
19. Zghal, S., Kachroudi, M., Damak, S.: Alignement d'ontologies base d'instances indexées. In: Actes de la 6$^{\text{èmes}}$ Edition des Journées Francophones sur les Ontologies (JFO'2016), Bordeaux, France (2016) 69–74

Lily Results for OAEI 2018

Yezhou Tang, Peng Wang*, Zhe Pan, Huan Liu

School of Computer Science and Engineering, Southeast University, China
`pwang@seu.edu.cn`

Abstract. This paper presents the results of Lily in the ontology alignment contest OAEI 2018. As a comprehensive ontology matching system, Lily is intended to participate in six tracks of the contest: conference, anatomy, largebio, phenotype, biodiv and spimbench. The specific techniques used by Lily will be introduced briefly. The strengths and weaknesses of Lily will also be discussed.

1 Presentation of the system

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

1.1 State, purpose, general statement

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

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

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

1.2 Specific techniques used

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

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

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

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

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

Matching weak informative ontologies Most existing ontology matching methods are based on the linguistic information. However, some ontologies may lack in regular linguistic information such as natural words and comments. Consequently the linguistic-based methods will not work. Structure-based methods are more practical for such situations. Similarity propagation is a feasible idea to realize the structure-based matching. But traditional propagation strategies do not take into consideration the ontology features and will be faced with effectiveness and performance problems. Having analyzed the classical similarity propagation algorithm, *Similarity Flood*, we proposed a new structure-based ontology matching method [5]. This method has two features: (1) It has more strict but reasonable propagation conditions which lead to more efficient matching processes and better alignments. (2) A series of propagation strategies are used to improve the matching quality. We have demonstrated that this method performs well on the OAEI benchmark dataset [5].

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

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

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

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

Currently, ontology matching tuning is not totally automatic. It is difficult to find out typical statistical parameters that distinguish each task from others.

Background Knowledge Matching Lily used matching strategy based on background knowledge this year. Lily has two sources of background knowledge:

the UMLS Metathesaurus, two synonyms files which contain a series of synonyms of many common medical terms and we obtain it via API of bioportal.com in advance. These two background knowledge sources are all specific to the biomedical domain such as largebio and phenotype track. Using background knowledge can greatly improve the matching effectiveness and efficiency to some extent. In the future, Lily will explore more effective background knowledge for other OAEI tracks or other matching tasks in the real world.

Virtual Document This year Lily used virtual document matching technology in some matching tasks[12]. Basically, as a collection of weighted words, the virtual document of a URIref declared in an ontology contains not only the local descriptions but also the neighboring information to reflect the intended meaning of the URIref. Document similarity can be computed by traditional vector space techniques, and then be used in the similarity-based approaches to ontology matching. Different matching tasks may have different neighbour information and weighted parameters to tune.

1.3 Adaptations made for the evaluation

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

1.4 Link to the system

SEALS wrapped version of Lily for OAEI 2018 is available at `https://drive.google.com/open?id=1irGjC4tZdofpG57kHXpblBJcf75ZwUWf`.

2 Results

2.1 Anatomy track

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

Table 1. The performance in the anatomy task

Matcher	Precision	Recall	Recall+	F-Measure
Lily	0.872	0.795	0.518	0.832

Compared with the result in OAEI 2016 [11], there is no obvious progress(with 0.83 F-Measure). As can be seen in the overall results, Lily lies in the middle position of the rank, which indicates that it is still possible to make further progress. Inside current Lily for anatomy, we used LOM(Large scale ontology matching) technique as mentioned in PART 1.2. In the future, we will add background knowledge into Lily for better matching result.

2.2 Conference track

Lily's performance in the Conference track was exactly the same as OAEI 2016. Obviously, Lily did not output satisfactory results in this track. The performance of Lily was even worse than StringEquiv in some tasks, which is a strange phenomenon. We will further analyze this task and our system to find out the reason later.

2.3 Disease and Phenotype track

Lily participated in this track for the first time. Lily generated almost the most unique mappings(733 in HP-MP task and 1167 in DOID-ORDO task).

Table 2. The performance in the disease and phenotype task

Matcher	Task	Mappings	Unique	Precision	Recall	F-Measure
Lily	HP-MP	2118	733	0.682	0.647	0.664
Lily	DOID-ORDO	3738	1167	0.589	0.783	0.672

However, Lily obtained a relatively low F-measure according to the 3-vote silver standard(0.664 and 0.672 separately). In our matching algorithm, we used classic virtual document technique and background knowledge matching strategy[12]. For the latter, we used a dictionary of synonyms extracted from BioPortal in advance. The reason why our precision is not high may be that the threshold of our virtual document was set too low, which caused many incorrect mappings. In addition, we think current consensus alignment(reference) using voting strategy is unreasonable to some extent for Lily. Since it may be not exactly the same as the gold matching results. For example, it perhaps missed some true mappings. However, these mappings are possible in unique mappings that Lily output but this voting strategy didn't count this part possibly, which led Lily to a low recall value relatively. Anyway, we will further optimize the algorithm inside Lily to make it cope with biological matching tasks better next year.

2.4 Biodiversity and Ecology track

Table 3. The performance in the biodiversity and ecology task

Matcher	Task	Precision	Recall	F-Measure
Lily	FLOPO-PTO	0.813	0.586	0.681
Lily	ENVO-SWEET	0.866	0.641	0.737

Lily obtained 68% F-measure in the FLOPO-PTO task and 73.7% F-measure in the ENVO-SWEET task. The results are not good because of low recall value relatively. In this task, we only considered simple text information(localName, label) for matching and ignored other potential information(structural information etc.). Consequently, Lily couldn't find more true mappings lacking of those information.

2.5 Spimbench track

This is an instance-mactching track which aims to match instances of creative works between two boxes. And ontology instances are described through 22 classes, 31 DatatypeProperty and 85 ObjectProperty properties.

There are about 380 instances and 10000 triples in sandbox, and about 1800 CWs and 50000 triples in mainbox.

Table 4. The performance in the spimbench task

Matcher	Scale	Precision	Recall	F-Measure
Lily	sandbox	0.8494	1.0000	0.9185
Lily	mainbox	0.8546	1.0000	0.9216

As is shown in Table 4, Lily utilized almost the same startegy to handle these two different size tasks. We found that creative works in this task were rich in text information such as titles, descriptions and so on. Lily could make good use of it and got the highest F-Measure with shortest time. However, garbled texts and messy codes were mixed up with normal texts. And Lily relied too much on text similarity calculation and set a low threshold in this task, which accounted for the low percision.

3 General comments

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

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

4 Conclusion

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

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

5 Acknowledgments

This work was supported by the National Natural Science Foundation of China (61472076 and 61472077).

References

[1] Peng Wang, Baowen Xu: Lily: ontology alignment results for OAEI 2009. In The 4th International Workshop on Ontology Matching, Washington Dc., USA (2009)

[2] Peng Wang, Baowen Xu: Lily: Ontology Alignment Results for OAEI 2008. In The Third International Workshop on Ontology Matching, Karlsruhe, Germany (2008)

[3] Peng Wang, Baowen Xu: LILY: the results for the ontology alignment contest OAEI 2007. In The Second International Workshop on Ontology Matching (OM2007), Busan, Korea (2007)

[4] Peng Wang, Baowen Xu, Yuming Zhou: Extracting Semantic Subgraphs to Capture the Real Meanings of Ontology Elements. Journal of Tsinghua Science and Technology, vol. 15(6), pp. 724-733 (2010)

[5] Peng Wang, Baowen Xu: An Effective Similarity Propagation Model for Matching Ontologies without Sufficient or Regular Linguistic Information, In The 4th Asian Semantic Web Conference (ASWC2009), Shanghai, China (2009)

[6] Peng Wang, Yuming Zhou, Baowen Xu: Matching Large Ontologies Based on Reduction Anchors. In The Twenty-Second International Joint Conference on Artificial Intelligence (IJCAI 2011), Barcelona, Catalonia, Spain (2011)

[7] Peng Wang, Baowen Xu: Debugging Ontology Mapping: A Static Approach. Computing and Informatics, vol. 27(1), pp. 2136 (2008)

[8] Peng Wang: Lily results on SEALS platform for OAEI 2011. Proc. of 6th OM Workshop, pp. 156-162 (2011)

[9] Yang, Pan, Peng Wang, Li Ji, Xingyu Chen, Kai Huang, Bin Yu: Ontology Matching Tuning Based on Particle Swarm Optimization: Preliminary Results. In The Semantic Web and Web Science, pp. 146-155 (2014)

[10] Peng Wang, Wenyu Wang: Lily results for OAEI 2015. In The 10th International Workshop on Ontology Matching, Bethlehem, PA, USA(2015)

[11] Peng Wang, Wenyu Wang: Lily results for OAEI 2016. In The 11th International Workshop on Ontology Matching, Kobe, Japan(2016)

[12] Qu Y, Hu W, Cheng G: Constructing virtual documents for ontology matching[C]// Proc. International Conference on World Wide Web. 23-31(2006)

LogMap family participation in the OAEI 2018

E. Jiménez-Ruiz[1,2], B. Cuenca Grau[3], and V. Cross[4]

[1] The Alan Turing Institute, London, UK
[2] Department of Informatics, University of Oslo, Oslo, Norway
[3] Department of Computer Science, University of Oxford, Oxford, UK
[4] Computer Science and Software Engineering, Miami University, Oxford, OH, United States

Abstract. We present the participation of LogMap and its variants in the OAEI 2018 campaign. The LogMap project started in January 2011 with the objective of developing a scalable and logic-based ontology matching system. This is our eight participation in the OAEI and the experience has so far been very positive. LogMap is one of the few systems that participates in (almost) all OAEI tracks.

1 Presentation of the system

LogMap [11, 13] is a highly scalable ontology matching system that implements the consistency and locality principles [12]. 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 [11, 13] for details).

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

Logic-based module extraction. The practical feasibility of unsatisfiability detection and repair critically depends on the size of the input ontologies. To reduce the size of the problem, we exploit ontology modularisation techniques. Ontology modules with well-understood semantic properties can be efficiently computed and are typically much smaller than the input ontology (e.g. [5]).

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

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

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

Semantic indexation. The Horn propositional representation of the ontology modules and the mappings is efficiently indexed using an interval labelling schema [1] — an optimised data structure for storing directed acyclic graphs (DAGs) that significantly reduces the cost of answering taxonomic queries [4, 19]. In particular, this semantic index allows us to answer many entailment queries as an index lookup operation over the input ontologies and the mappings computed thus far, and hence without the need for reasoning. The semantic index complements the use of the propositional encoding to detect and repair unsatisfiable classes.

1.1 LogMap variants in the 2018 campaign

As in previous campaigns, in the OAEI 2018 we have participated with two additional variants:

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

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

In previous years we also participated with LogMapC[5].

1.2 Adaptations made for the 2018 evaluation

LogMap's algorithm described in [11, 13, 16, 15, 14] has been adapted with the following new functionalities:

i **HOBBIT adaptation**. We have implemented the required interface classes to run LogMap under the HOBBIT platform.[6] LogMap can currently be evaluated in five different tracks available in the HOBBIT platform.[7]

ii **Ontology division module**. This module extends LogMap's ontology overlapping estimation module to compute a number of divisions of the input ontologies and to create a set of smaller matching subtasks [10].

iii **Obsolete classes**. We have extended the lexical and structural indexation modules to ignore classes in the ontology annotated as obsolete.

1.3 Link to the system and parameters file

LogMap is open-source and released under GNU Lesser General Public License 3.0.[8] LogMap components and source code are available from the LogMap's GitHub page: `https://github.com/ernestojimenezruiz/logmap-matcher/`.

---

[5] LogMapC is a variant of LogMap which, in addition to the consistency and locality principles, also implements the conservativity principle (see details in [20–22, 18]).

[6] `https://gitlab.com/ernesto.jimenez.ruiz/logmap-hobbit`

[7] `https://git.project-hobbit.eu/ernestoj/logmapsystem`

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

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://krrwebtools.cs.ox.ac.uk/`. This interface has been very well received by the community since it was deployed in 2012. More than 3,000 requests coming from a broad range of users have been processed so far.

1.4 LogMap as a mapping repair system

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

We believe that these systems could significantly improve their output if they were to implement repair techniques similar to those available in LogMap. Therefore, with the goal of providing a useful service to the community, we have made LogMap's ontology repair module (LogMap-Repair) available as a self-contained software component that can be seamlessly integrated in most existing ontology matching systems [17, 7].

1.5 LogMap as a matching task division system

LogMap also includes a novel module to divide the ontology alignment task into (independent) manageable subtasks [10]. This component relies on LogMap's lexical index, a neural embedding model [23] and locality-based modules [5]. This module can be integrated in existing ontology alignment systems as a external module. The preliminaty results in [10] are encouraging as the division enabled systems to complete some large-scale matching tasks.

2 General comments and conclusions

Please refer to `http://oaei.ontologymatching.org/2018/results/` for the results of the LogMap family in the OAEI 2018 campaign.

2.1 Comments on the results

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

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

Acknowledgements

This work was partially supported by the AIDA project, funded by the UK Government's Defence & Security Programme in support of the Alan Turing Institute, and the SIRIUS Centre for Scalable Data Access (Research Council of Norway, project no.: 237889).

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

References

1. Agrawal, R., Borgida, A., Jagadish, H.V.: Efficient management of transitive relationships in large data and knowledge bases. In: ACM SIGMOD Conf. on Management of Data. pp. 253–262 (1989)
2. Baeza-Yates, R.A., Ribeiro-Neto, B.A.: Modern Information Retrieval. ACM Press / Addison-Wesley (1999)
3. Chen, X., Xia, W., Jiménez-Ruiz, E., Cross, V.: Extending an ontology alignment system with bioportal: a preliminary analysis. In: Poster at Int'l Sem. Web Conf. (ISWC) (2014)
4. Christophides, V., Plexousakis, D., Scholl, M., Tourtounis, S.: On labeling schemes for the Semantic Web. In: Int'l World Wide Web (WWW) Conf. pp. 544–555 (2003)
5. Cuenca Grau, B., Horrocks, I., Kazakov, Y., Sattler, U.: Modular reuse of ontologies: Theory and practice. J. Artif. Intell. Res. 31, 273–318 (2008)
6. Dowling, W.F., Gallier, J.H.: Linear-time algorithms for testing the satisfiability of propositional Horn formulae. J. Log. Prog. 1(3), 267–284 (1984)
7. Faria, D., Jiménez-Ruiz, E., Pesquita, C., Santos, E., Couto, F.M.: Towards annotating potential incoherences in bioportal mappings. In: 13th Int'l Sem. Web Conf. (ISWC) (2014)
8. Fridman Noy, N., Shah, N.H., Whetzel, P.L., Dai, B., et al.: BioPortal: ontologies and integrated data resources at the click of a mouse. Nucleic Acids Research 37, 170–173 (2009)
9. Ghazvinian, A., Noy, N.F., Jonquet, C., Shah, N.H., Musen, M.A.: What four million mappings can tell you about two hundred ontologies. In: Int'l Sem. Web Conf. (ISWC) (2009)
10. Jiménez-Ruiz, E., Agibetov, A., Samwald, M., Cross, V.: Breaking-down the ontology alignment task with a lexical index and neural embeddings. CoRR abs/1805.12402 (2018), `http://arxiv.org/abs/1805.12402`
11. Jiménez-Ruiz, E., Cuenca Grau, B.: LogMap: Logic-based and Scalable Ontology Matching. In: Int'l Sem. Web Conf. (ISWC). pp. 273–288 (2011)
12. Jiménez-Ruiz, E., Cuenca Grau, B., Horrocks, I., Berlanga, R.: Logic-based assessment of the compatibility of UMLS ontology sources. J. Biomed. Sem. 2 (2011)
13. Jiménez-Ruiz, E., Cuenca Grau, B., Zhou, Y., Horrocks, I.: Large-scale interactive ontology matching: Algorithms and implementation. In: Europ. Conf. on Artif. Intell. (ECAI) (2012)
14. Jiménez-Ruiz, E., Grau, B.C., Cross, V.: Logmap family participation in the OAEI 2017. In: Proceedings of the 12th International Workshop on Ontology Matching co-located with the 16th International Semantic Web Conference (ISWC 2017), Vienna, Austria, October 21, 2017. pp. 153–157 (2017), `http://ceur-ws.org/Vol-2032/oaei17\_paper7.pdf`
15. Jiménez-Ruiz, E., Grau, B.C., Cross, V.V.: Logmap family participation in the OAEI 2016. In: Proceedings of the 11th International Workshop on Ontology Matching co-located with the 15th International Semantic Web Conference (ISWC 2016), Kobe, Japan, October 18, 2016. pp. 185–189 (2016), `http://ceur-ws.org/Vol-1766/oaei16_paper9.pdf`

16. Jiménez-Ruiz, E., Grau, B.C., Solimando, A., Cross, V.V.: Logmap family results for OAEI 2015. In: Proceedings of the 10th International Workshop on Ontology Matching collocated with the 14th International Semantic Web Conference (ISWC 2015), Bethlehem, PA, USA, October 12, 2015. pp. 171–175 (2015), `http://ceur-ws.org/Vol-1545/oaei15_paper10.pdf`
17. Jiménez-Ruiz, E., Meilicke, C., Cuenca Grau, B., Horrocks, I.: Evaluating mapping repair systems with large biomedical ontologies. In: 26th Description Logics Workshop (2013)
18. Jimenez-Ruiz, E., Payne, T.R., Solimando, A., Tamma, V.: Limiting logical violations in ontology alignment through negotiation. In: Proceedings of the 15th International Conference on Principles of Knowledge Representation and Reasoning (KR). AAAI Press (April 2016)
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., Jimenez-Ruiz, E., Guerrini, G.: Minimizing conservativity violations in ontology alignments: Algorithms and evaluation. Knowledge and Information Systems (2016), `https://github.com/asolimando/logmap-conservativity/`
23. Wu, L., Fisch, A., Chopra, S., Adams, K., Bordes, A., Weston, J.: Starspace: Embed all the things! arXiv preprint arXiv:1709.03856 (2017)

OAEI 2018 results of POMap++

Amir Laadhar[1], Faiza Ghozzi[2], Imen Megdiche[1], Franck Ravat[1], Olivier Teste[1], and Faiez Gargouri[2]

[1] University of Toulouse, IRIT (CNRS/UMR 5505) 118 Route de Narbonne 31062
Toulouse, France
{`amir.laadhar,imen.megdiche,franck.ravat,olivier.teste`}`@irit.fr`,
[2] University of Sfax, MIRACL Sakiet Ezzit 3021, Tunisie
{faiza.ghozzi,faiez.gargouri}@isims.usf.tn

Abstract. Ontology matching is the process of finding a set of correspondences between the entities of two or more ontologies representing a similar domain. POMap++ is an ontology matching system associating ontology partitioning to the machine learning techniques. This association delivers a local matching learning. POMap++ provides an automated local matching learning for the biomedical tracks. For the non-biomedical tracks we employ the version of POMap 2017. In this paper, we present POMap++ as well as the obtained results for the Ontology Alignment Evaluation Initiative of 2018.

Keywords: Semantic web, Ontology Matching, Ontology partitioning, Machine learning

1 Presentation of the system

Ontologies are the backbone of the semantic web. They enable sharing, reusing and accessing the knowledge resources [9]. Biomedical ontologies are domain-specific knowledge bases widely employed in biology and medicine. These ontologies have been separately developed by different experts using different terminologies and modeling techniques. The integration of these data sources requires ontology matching tools. Ontology matching is the identification process correspondences between the entities of different ontologies. The alignment process is quite challenging in terms of the complexity of the existing biomedical ontologies. POMap++ divide a biomedical ontology alignment to a set of sub-matching tasks called partitions. We align each sub-matching task using its local adequate settings. We automatically determine the local matching settings by generating a specific machine learning model for each sub-matching task. This automated tuning process of local matching parameters aims to improve the overall matching quality of a large ontology matching task. We employed POMap++ for the biomedical matching tasks and POMap [3] for the non-biomedical matching tasks. In the following section, we provide a detailed description of POMap++.

1.1 State, purpose, general statement

1.2 Specific techniques used

The workflow of POMap++ for our second participation in the OAEI comprises four main steps, as flagged by the figure 1: Input ontologies indexing and loading, input ontologies partitioning, local matching learning and output alignment generation. The first and the last step are the same as in the last version of POMap [3]. In the second step, we define the pair of similar partitions between the two input ontologies. In the third step, we apply machine learning techniques in order to align every identified pair of similar partitions. In the following, we detail each of the four steps.

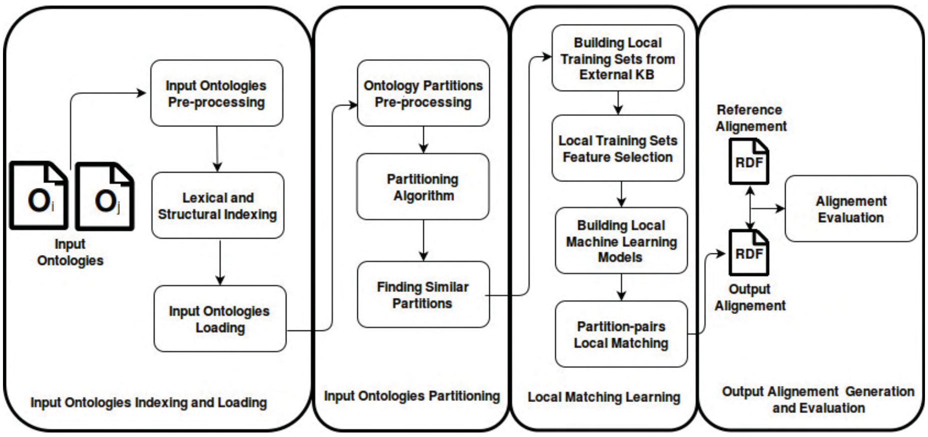


Fig. 1. The architecture of POMap++.

Step 1: Input ontologies indexing and loading

The first step of the ontology indexation and loading is the pre-processing task. We pre-process the annotations of the two input ontologies by applying the Porter stemming [8] as well as the stop word removal process. We also remove the special characters. These indexes are stored along with the structure of the input ontologies. The structural indexing is responsible for representing the relationships between entities. Then, during the third task, the indexed data structures are loaded into the next step of POMap++.

Step 2: Input ontologies partitioning

We divide an ontology into a set of partitions using the hierarchical agglomerative clustering [5] approach. This approach does not take as input the required number of partitions. The hierarchical agglomerative clustering algorithm receives as input structural similarity scores between all the entities of an input ontology. We compute the structural similarity between the entities of a single ontology according to the following Definition. The Definition 1 is inspired by Wu and Palmer [10] similarity measure.

Definition 1 (Structural similarity between entities). *We compute the structural similarity between all the entities in one ontology according to the Equation 1. For a given two entities* $e_{i,x}$ *and* $e_{i,y}$ *of an ontology* O_i*, lca is their lowest common ancestor. Dist(*$e_{i,x}$*,lca) represents the shortest distance between* $e_{i,x}$ *and lca in terms of number of edges. Dist(*$e_{i,y}$*,lca) denote the distance between* $e_{i,y}$ *and lca. Dist(*r_i*,lca) is the distance between the root* r_i *and lca.*

$$StrcSim(e_{i,x}, e_{i,y}) = \frac{Dist(r_i, lca) \times 2}{Dist(e_{i,x}, lca) + Dist(e_{i,y}, lca) + Dist(r_i, lca) \times 2} \quad (1)$$

Step 3: Local Matching learning

Due to the high complexity of biomedical ontologies, no single syntactic similarity measure can effectively all the syntactic heterogeneity of a matching task. Therefore, for each local matching task, we construct its specific machine learning model. The training set of every local learning model is not based on any reference alignments. We automatically construct a supervised training set for each local matching task of the set of local matchings. These training sets serve as the input for each local machine learning model. After identifying the partitions for each ontology, we find the set of similar partitions between the two input ontologies using a set of anchors. The existing works retrieve labeled data either from the reference alignment or by creating it manually. However, the reference alignment commonly does not exist. We derive each local training set by cross-searching the entities of a local matching with the existing biomedical knowledge bases like Uberon. Since we are dealing with biomedical ontologies, anchors are extracted by cross-searching the input ontologies with the available external biomedical knowledge bases (KB) such as the Unified Medical Language System (UMLS) Metathesaurus [1], Medical Subject Headings (MeSH) [4], Uberon [6] and BioPortal [7]. For instance, UMLS integrates more than 160 biomedical ontologies. In our case, we cross-search the two input ontologies with the Uberon ontology to derive the set anchors. We employ the-state-of-the art syntactic similarity measures[3] as features. The labeled data of the training set is usually hard to acquire. We apply the wrapper feature selection [2] method over the resulted local training sets. This technique selects the subset of the most effective and suitable features for each local training set. Therefore, each local matching task has its specific similarity measures. Then, we build a local machine learning model for each local matching task. The entities of each local matching task are classified using their specific machine learning model. This local learning model aligns the input entities based on the adequate matching parameters.

Step 4: Output alignment generation

---

[3] https://git.io/fNvqt

The generated correspondences for every local matching task $lm_{ij,q}$ are unified to generate the final alignment file for the whole ontology matching task. The alignment file is compared to the reference alignment to evaluate the overall result accuracy.

2 Results

2.1 Anatomy

The Anatomy track consists of finding the alignments between the Adult Mouse Anatomy and the NCI Thesaurus describing the human anatomy. The evaluation was run on a server coupled with 3.46 GHz (6 cores) and 8GB of RAM. Table 1 draws the performance of POMap++ compared to the five top matching systems. Our matching system achieved the third best result for this dataset with an F-measure of 89.7%, which is very close to the top results. The remaining challenge is to speed up the execution time by applying more optimizations. We also target the improvement of precision value for our next participation in the OAEI.

2.2 Disease and Phenotype

This track is based on a real use case in order to find alignments between disease and phenotype ontologies. Specifically, the selected ontologies are the Human Phenotype Ontology (HPO), the Mammalian Phenotype Ontology (MP), the Human Disease Ontology (DOID) and the Orphanet and Rare Diseases Ontology(ORDO). The evaluation was run on an Ubuntu Laptop with an Intel Core i9-8950UK CPU @ 2.90GHz x 12 coupled with 25Gb RAM. POMap++ succeeded to complete tow tasks HP-MP and DOID-ORDO. POMap produced 1502 mappings in the HP-MP task associated with 214 unique mappings. Among the eight matching systems, POMap++ achieved the fifth highest F-measure with an F-Measure of 69.9%. In the DOID-ORDO task, POMap generated 2563 mappings with 174 unique ones. For this task, POMap++ obtained an F-Measure of 84.5% being the third best result for this track.

2.3 LargeBio

This tracks aims to find the alignment between three large ontologies: Foundational Model of Anatomy (FMA), SNOMED CT, and the National Cancer Institute Thesaurus (NCI). Among six matching tasks between these three ontologies, POMap++ succeeded to perform the matching between FMA-NCI (small fragments) and FMA-SNOMED (small fragments) with an F-Measure respectively of 88.9% and 40.4%. For the other tasks of the large biomedical track, POMap++ exceeded the defined timeout.

3 Conclusion

The obtained results of POMap++ are promising especially for disease and phenotype as well as the anatomy track in which we ranked as the third top performing matching system. However, we did not opt to perform the local matching using structural-level features. Consequently, we are planning to add structural-level feature to the local matching process.

References

1. Bodenreider, O.: The unified medical language system (umls): integrating biomedical terminology. vol. 32, pp. D267–D270. Oxford University Press (2004)
2. Kohavi, R., John, G.H.: Wrappers for feature subset selection. vol. 97, pp. 273–324. Elsevier (1997)
3. Laadhar, A., Ghozzi, F., Megdiche, I., Ravat, F., Teste, O., Gargouri, F.: Pomap results for oaei 2017. In: 12th International Workshop on Ontology Matching collocated with the 16th International Semantic Web Conference (OM@ ISWC'17). pp. pp–1 (2017)
4. Lipscomb, C.E.: Medical subject headings (mesh). vol. 88, p. 265. Medical Library Association (2000)
5. Müllner, D.: Modern hierarchical, agglomerative clustering algorithms (2011)
6. Mungall, C.J., Torniai, C., Gkoutos, G.V., Lewis, S.E., Haendel, M.A.: Uberon, an integrative multi-species anatomy ontology. vol. 13, p. R5. BioMed Central (2012)
7. Noy, N.F., Shah, N.H., Whetzel, P.L., Dai, B., Dorf, M., Griffith, N., Jonquet, C., Rubin, D.L., Storey, M.A., Chute, C.G., et al.: Bioportal: ontologies and integrated data resources at the click of a mouse. vol. 37, pp. W170–W173. Oxford University Press (2009)
8. Porter, M.F.: Snowball: A language for stemming algorithms (2001)
9. Shvaiko, P., Euzenat, J.: Ontology matching: state of the art and future challenges. IEEE Transactions on knowledge and data engineering 25(1), 158–176 (2013)
10. Wu, Z., Palmer, M.: Verbs semantics and lexical selection. In: Proceedings of the 32nd annual meeting on Association for Computational Linguistics. pp. 133–138. Association for Computational Linguistics (1994)

RADON2: A buffered-Intersection Matrix Computing Approach To Accelerate Link Discovery Over Geo-Spatial RDF Knowledge Bases

OAEI2018 Results

Abdullah Fathi Ahmed[1] Mohamed Ahmed Sherif[1,2] and Axel-Cyrille Ngonga Ngomo[1,2]

[1] Paderborn University, Data Science Group, Pohlweg 51, D-33098 Paderborn, Germany
E-mail: {firstname.lastname}@upb.de
[2] Department of Computer Science, University of Leipzig, 04109 Leipzig, Germany
E-mail: {lastname}@informatik.uni-leipzig.de

Abstract. Geospatial data is at the essence of the Semantic Web, where a knowledge base such as *LinkedGeoData* consists of more than 30 billions facts. Reasoning on these considerable amounts of geospatial data lacks efficient methods for the computation of links between the resources contained in these knowledge bases. In this paper, we present the participation of the extension of Radon algorithm (dubbed Radon2) in the OAEI 2018 campaign. The OAEI results show that Radon2 outperforms the other state of the art in most of the cases.

1 Presentation of the System

we present the extension of Radon algorithm [8, 6] (dubbed Radon2), where we, compute all topological relations of DE9-IM in order to accelerate the topological relation discovery among geospatial resources.

1.1 State, Purpose and General Statement

In the following, we start by formally defining the general link discovery problem. Thereafter, we formally define the link discovery of topological relations problem, which we takeld by Radon2.

Link Discovery. Let K be a finite RDF knowledge base. K can be regarded as a set of triples $(s, p, o) \in (\mathcal{R} \cup \mathcal{B}) \times \mathcal{P} \times (\mathcal{R} \cup \mathcal{L} \cup \mathcal{B})$, where $\mathcal{R}$ is the set of all resources, $\mathcal{B}$ is the set of all blank nodes, $\mathcal{P}$ the set of all predicates and $\mathcal{L}$ the set of all literals. The Link Discovery (LD) problem can be expressed as follows: Given two sets of resources S and T (for example hotels and water bodies) and a relation r (e.g., `:touches`), find all pairs $(s, t) \in S \times T$ such that $r(s, t)$ holds. The result is produced as a set of links called a *mapping*: $M_{S,T} = \{(s_i, r, t_j) | s_i \in S, t_j \in T\}$. Optionally, a similarity score ($sim \in [0, 1]$) calculated by an LD tool can be added to the entries of mappings to express assurance of a computed link. Finding solutions for the LD problem is challenging due to the typically the large volume of current datasets as well as its semantic heterogeneity. The main purpose of LD approaches is to meet the main requirements of *(1) high effectiveness* (i.e maximize a fitness function such as F-measure) and *(2) high efficiency* (i.e., minimize runtime).

Link Discovery of Topological Relations. The Dimensionally Extended nine-Intersection Model (DE-9IM) [3] is a topological model and a standard used to describe the spatial relations of two geometries in two-dimensional space. Since the spatial relations expressed by DE-9IM are topological, they are invariant to rotation, translation and scaling transformations [4]. The DE-9IM model is based on a 3×3 intersection matrix with the form:

$$DE9IM(g_1, g_2) = \begin{bmatrix} dim(I(g_1) \cap I(g_2)) & dim(I(g_1) \cap B(g_2)) & dim(I(g_1) \cap E(g_2)) \\ dim(B(g_1) \cap I(g_2)) & dim(B(g_1) \cap B(g_2)) & dim(B(g_1) \cap E(g_2)) \\ dim(E(g_1) \cap I(g_2)) & dim(E(g_1) \cap B(g_2)) & dim(E(g_1) \cap E(g_2)) \end{bmatrix} \quad (1)$$

where dim is the maximum number of dimensions of the intersection $\cap$ of the $interior(I)$, $boundary(B)$, or $exterior(E)$ of the two geometries g_1 and g_2. The domain of dim is $\{-1, 0, 1, 2\}$, where -1 indicates no intersection, 0 stands for an intersection that results in a set of one or more points, 1 indicates an intersection made up of lines and 2 stands for an intersection that results in an area. A simplified binary version of $dim(x)$ with the binary domain $\{true, false\}$ is obtained using the Boolean function $\beta(dim(I(g)) = false$ iff $dim(I(g)) = -1$ and $true$ otherwise. There is only a subset of the topological relations obtainable through DE-9IM that reflects the semantics of the English language [3] [2] including `equals, within, contains, disjoint, touches, meets, covers, coveredBy, intersects, crosses` and `overlaps`.

1.2 Specific Techniques Used

in this section, we discuss the main idea behind our new extension of Radon.

Radon2 *vs.* Radon*.* The basic idea behind the original Radon approach [8] for topological relation discovery is to provide an indexing method combined with space tiling that allows for efficient computation of topological relations between geospatial resources. In particular, Radon presents a novel sparse index for geospatial resources. Then, based on bounding boxes of the indexed geospatial resources, Radon applies a strategy for discarding unnecessary computations of DE-9IM relations. In Radon2, our concerns is focused on optimizing the computing of intersection matrix (IM) used in DE9-IM standard. In the original Radon, the intersection matrix is computed for each topological relation, while in Radon2 we compute the IM once for all relations among the same pair of resources. We then apply the mask for each relation to the the computed IM. In particular, we buffer the IM of each pair of geometries so that all topological relations of same pair can be retrieved with no need to recompute their respective IM again. By applying this strategy, we can save the time for recomputing the IM for each individual topological relation. Moreover, calculating IM at once for each pair of geometries for all topological relations does not affect the completeness of the linking result. i.e., the F-measure of Radon2 is the same as the F-measure of Radon, which is always 1.

1.3 Adaptations Made for the Evaluation

No specific adaptations were made to the original Radon algorithm, we only provide a Java `SystemAdapter` according to the campaign guidelines[3].

---

[3] https://project-hobbit.eu/challenges/om2017/om2017-tasks/

1.4 Link to the System

Both Radon and Radon2 are implemented in the link discovery framework Limes. Limes is available under the *GNU Affero General Public License v3.0*[4]. Radon2 source code is available online from the project website[5]. The project web site also provide a user manual[6] as well as a developer manual[7].

2 Results

Radon2 has been evaluated only in the Hobbit Link Discovery Track Task 2 (Spatial). The basic idea behind this task was to measure how well the systems can identify DE- 9IM (Dimensionally Extended nine-Intersection Model) topological relations. The supported spatial relations were: `equals, within, contains, disjoint, touches, meets, covers, coveredBy, intersects, crosses` and `overlaps`. The geospatial resources traces were represented in Well-known text (WKT) format as LineStrings. The result is produced as a set of links called a *mapping*: $M_{S,T} = \{(s_i, r, t_j) | s_i \in S, t_j \in T\}$. All the systems were tested against two datasets: (1) the sandbox dataset, with a scale of 10 instances, and (2) the mainbox dataset with a scale of $5K$ instances. The other participants to this task in addition to Radon were Agreement Maker Light(AML) and Silk.

The systems were judged on the basis of precision, recall, F-Measure and run time. The final results are shown in Figures 1, 2, 3 and 4. Note that we are only presenting the time performance and not precision, recall and F-Measure as all were equal to 1.0.

From these results we can see that Radon2 outperforms the other systems in all relations for the sandbox and mainbox (linestrings –polygons) (see Figures 3 and 4) dataset as well as the for the the mainbox dataset (linestrings–linestrings) (Figure 2). For the sandbox dataset (linestrings–linestrings) (Figure 1), Radon achives a better performance in most of the relations (e.g., `overlaps, crosses, covered by, covers, within, contains, disjoint and equal`. Only for the `touches and intersects` Aml was able to outperform Radon2 for the TomTom dataset of the sandbox (linestrings–linestrings). The differences in performance between `touches` and `intersects`, where AML outperforms Radon cannot be explained from an implementation point of view, as these two relations share the exact optimizations. However, due to the datasets consisting exclusively of `LineStrings`, it is apparent that *touches* and *intersects* are much more likely to hold between any two geometries than other relations. Therefore, the benchmarks on these relations are the hardest in this task.

3 Conclusions and Future Work

We present Radon2, a simple strategy for scaling the original Radon approach by computing the intersection matrix for each pair of resources once and use it for computing all possible topological relations associated with such resources at hand. The presented evaluation during the OAEI 2018 showed that, in addition to being complete and correct (i.e. achieving an F-Measure of 1.0), Radon2 also outperforms the other participating systems in most of the cases

---

[4] https://github.com/dice-group/LIMES/blob/master/LICENSE

[5] https://github.com/dice-group/LIMES

[6] https://dice-group.github.io/LIMES/user_manual/

[7] https://dice-group.github.io/LIMES/developer_manual/

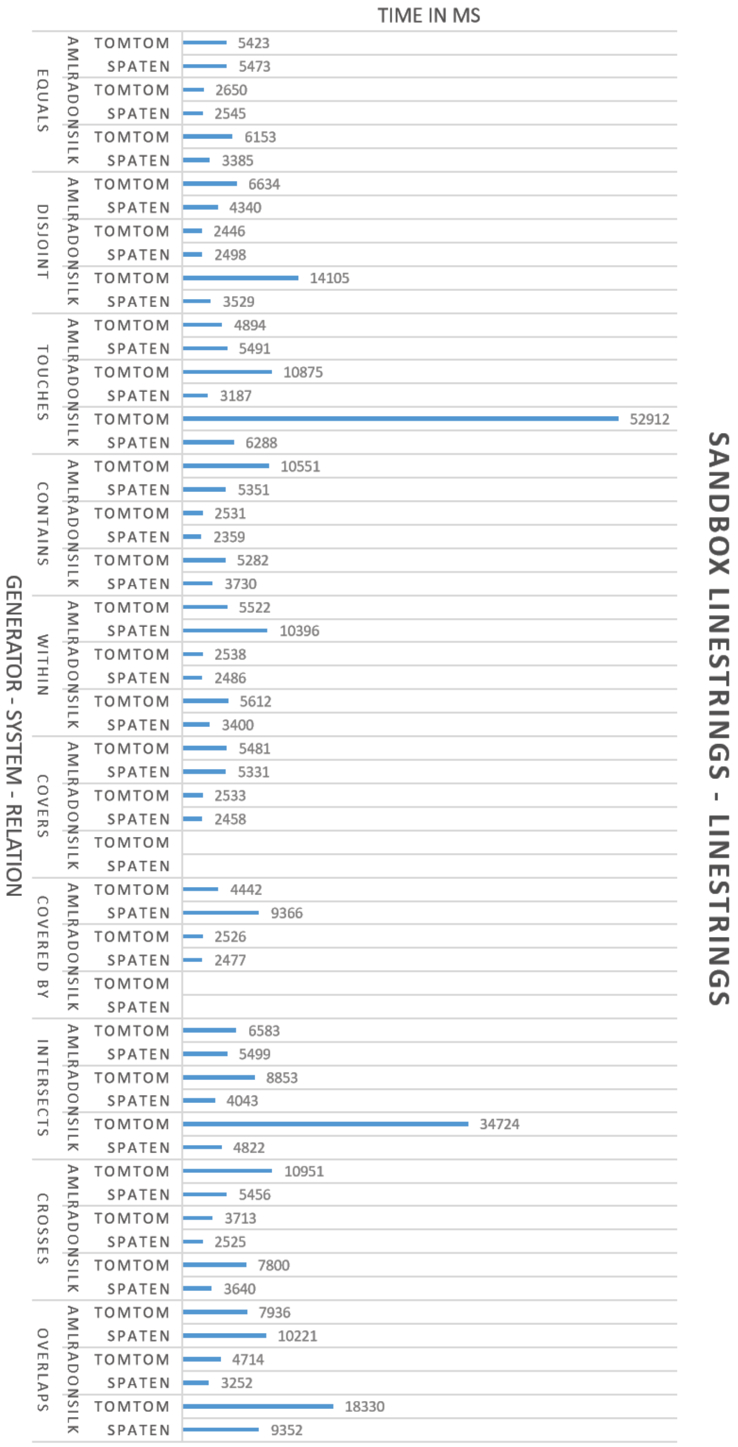


Fig. 1: Runtime results of linestrings-linestrings *Sandbox* Dataset

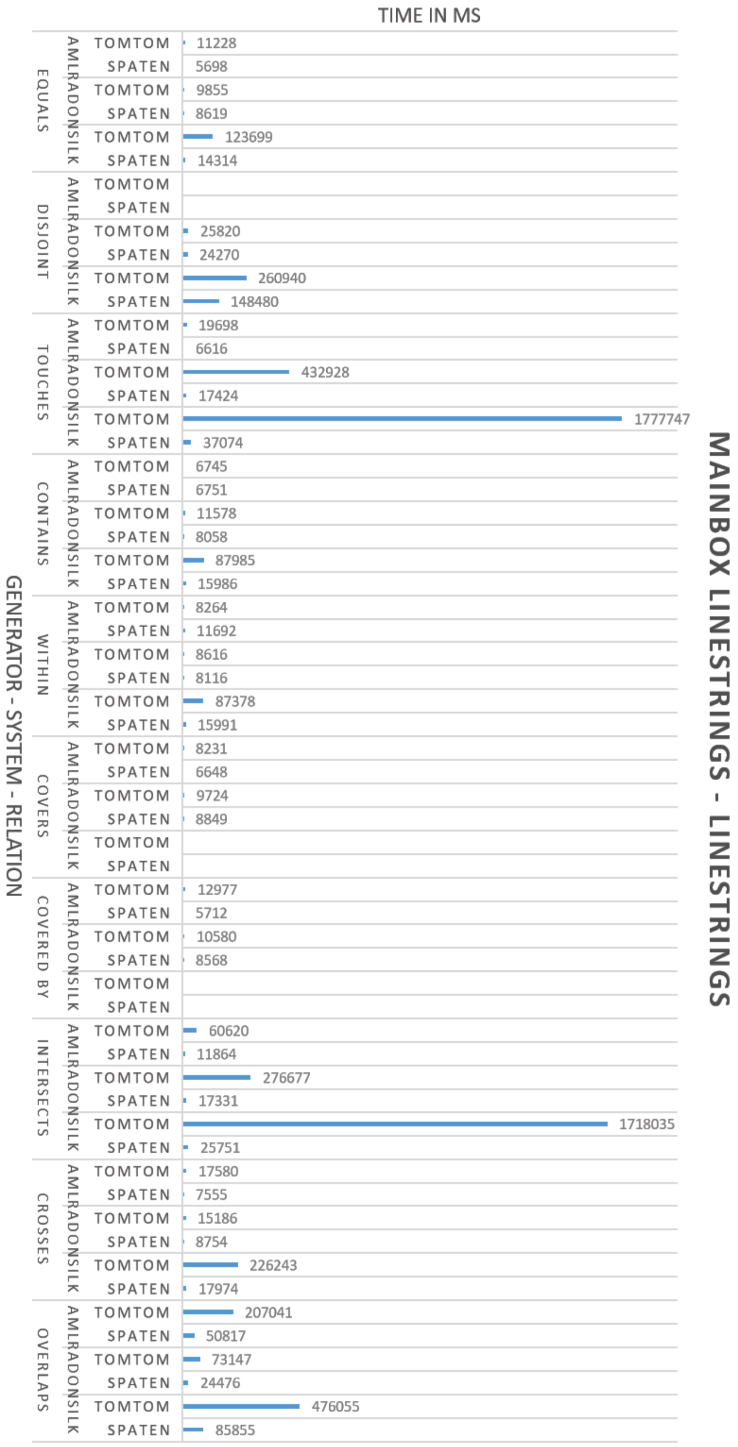


Fig. 2: Runtime results of linestrings-linestrings *Mailbox* DataSet

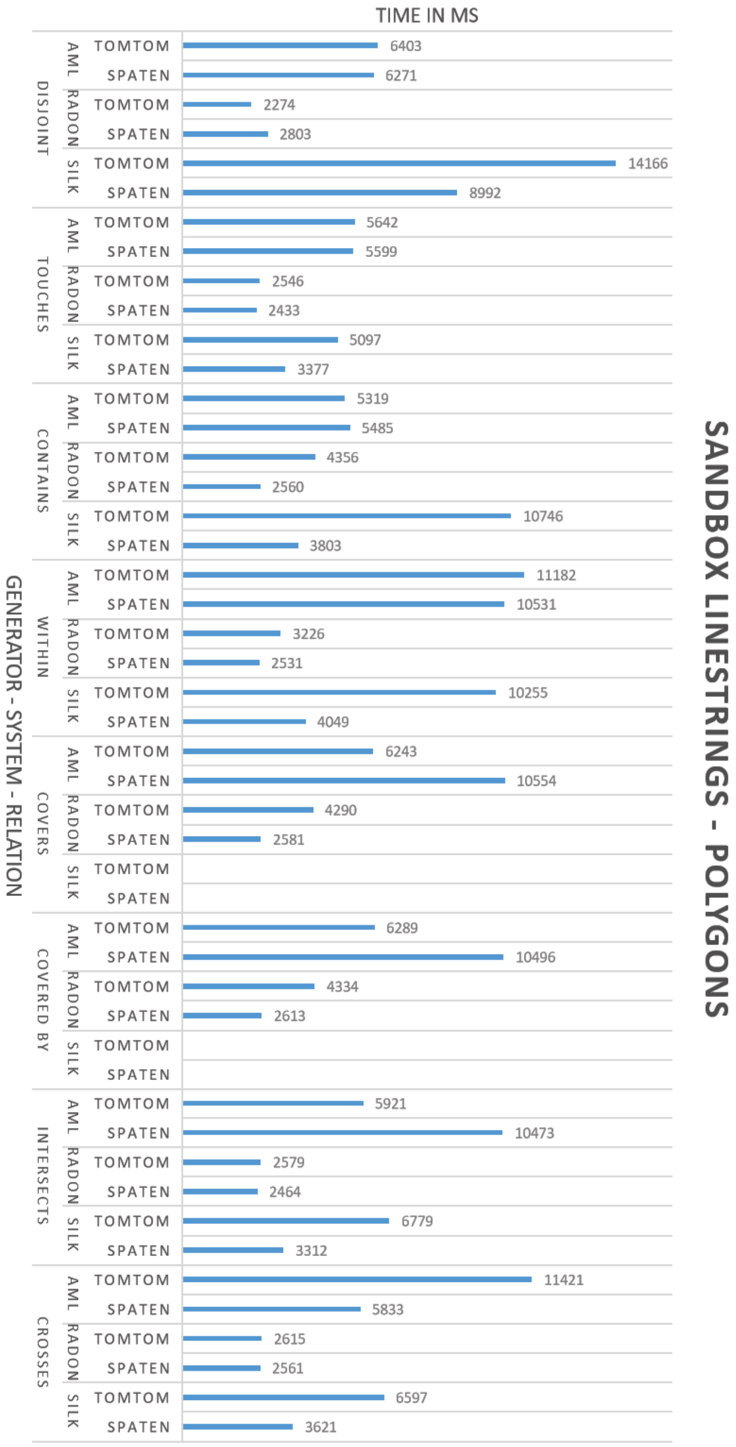


Fig. 3: Runtime results of linestrings-polygons *Sandbox* Dataset

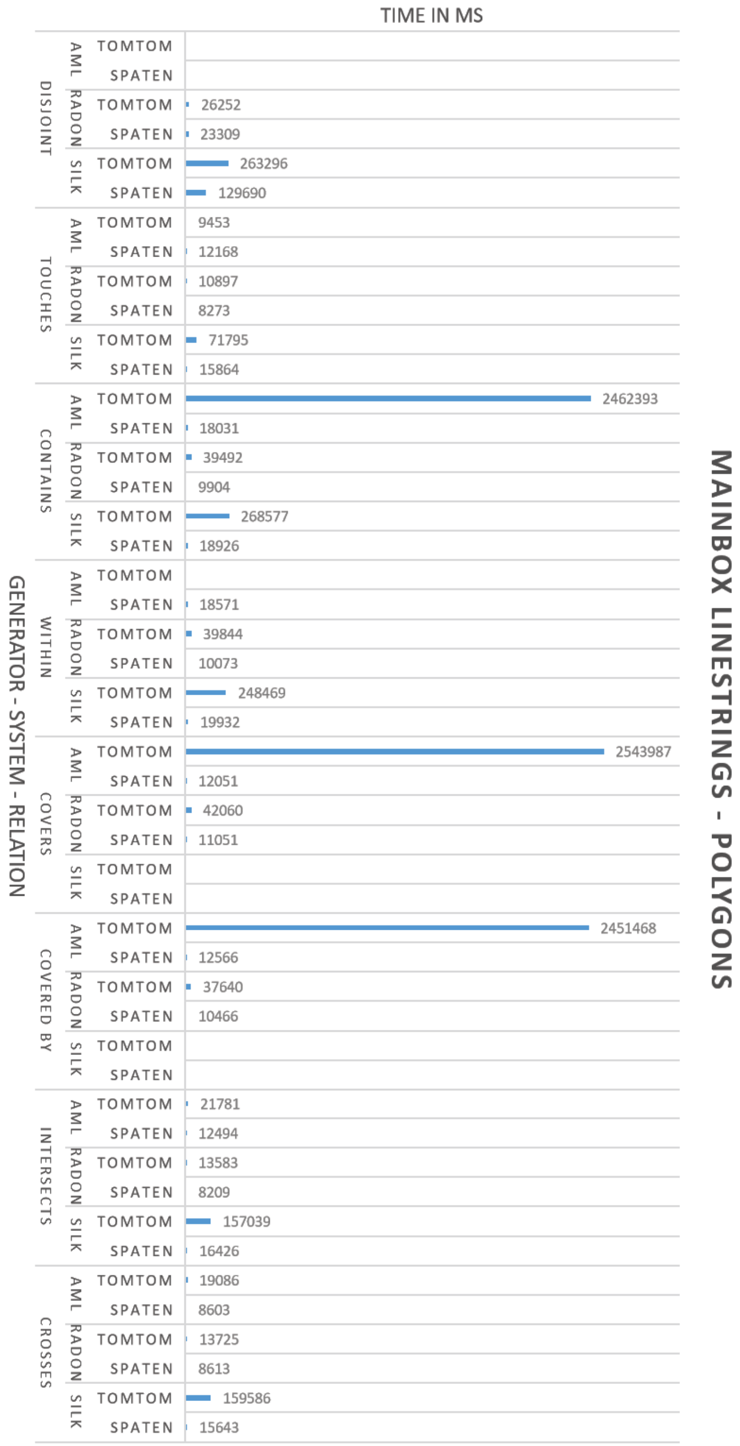


Fig. 4: Runtimes results of linestrings-polygons *Mailbox* DataSet

In future work, we will apply this strategy on a larger datasets with more resources and more points per resource, where we will implement more sophisticated parallelization techniques. For enabling automatic configuration of Radon2, we will combine Radon2 with the machine learning algorithm Wombat [7] implemented in Limes. Also, we will extend Radon2 for discovering spatial-temporal relation by integrating it with [5]. Moreover, we intend to combine Radon2 with the simplification algorithms introduced in [1] in order to achieve even better speedup.

Acknowledgments This work has been supported by Eurostars Project SAGE (GA no. E!10882), the BMVI project the LIMBO (GA no. 19F2029C), the DFG project LinkingLOD (project no. NG 105/3-2), the BMWI Project GEISER (project no. 01MD16014) as well as the H2020 projects SLIPO (GA no. 731581) and HOBBIT (GA no. 688227).

References

1. A. F. Ahmed, M. Sherif, and A. Ngonga Ngomo. On the effect of geometries simplification on geo-spatial link discovery. In *Proceedings of SEMANTiCS 2018*, 2018.
2. E. Clementini, P. Di Felice, and P. Van Oosterom. A small set of formal topological relationships suitable for end-user interaction. In *International Symposium on Spatial Databases*, pages 277–295. Springer, 1993.
3. E. Clementini, J. Sharma, and M. J. Egenhofer. Modelling topological spatial relations: Strategies for query processing. *Computers & graphics*, 18(6):815–822, 1994.
4. M. J. Egenhofer and R. D. Franzosa. Point-set topological spatial relations. *International Journal of Geographical Information System*, 5(2):161–174, 1991.
5. K. Jha, M. Röder, and A.-C. Ngonga Ngomo. All That Glitters is not Gold – Rule-Based Curation of Reference Datasets for Named Entity Recognition and Entity Linking. In *The Semantic Web. Latest Advances and New Domains: 14th International Conference, ESWC 2017, Proceedings*. Springer International Publishing, 2017.
6. M. A. S. a.-C. N. Kevin Dreßler. Radon results for oaei 2017. In *Proceedings of Ontology Matching Workshop 2017*, 2017.
7. M. Sherif, A.-C. Ngonga Ngomo, and J. Lehmann. WOMBAT - A Generalization Approach for Automatic Link Discovery. In *14th Extended Semantic Web Conference, Portorož, Slovenia, 28th May - 1st June 2017*. Springer, 2017.
8. M. A. Sherif, K. Dreßler, P. Smeros, and A.-C. Ngonga Ngomo. RADON - Rapid Discovery of Topological Relations. In *Proceedings of The Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)*, 2017.

SANOM Results for OAEI 2018

Majid Mohammadi, Wout Hofman, and Yao-Hua Tan

Faculty of Technology, Policy and Management, Delft University of Technology, The Netherlands,
TNO Research institute, The Netherlands.

Abstract. Simulated annealing-based ontology matching (SANOM) participates for the second time at the ontology alignment evaluation initiative (OAEI) 2018. This paper contains the configuration of SANOM and its results on the anatomy and conference tracks. In comparison to the OAEI 2017, SANOM has improved significantly, and its results are competitive with the state-of-the-art systems. In particular, SANOM has the highest recall rate among the participated systems in the conference track, and is competitive with AML, the best performing system, in terms of F-measure. SANOM is also competitive with LogMap on the anatomy track, which is the best performing system in this track with no usage of particular biomedical background knowledge. SANOM has been adapted to the HOBBIT platfrom and is now available for the registered users. *abstract* environment.

Keywords: SANOM, ontology alignment, OAEI.

1 System Representation

SANOM takes advantages of the well-known simulated annealing (SA) to discover the shared concepts between two given ontologies [3]. A potential alignment is modeled as a state in the SA whose evolution would result in a more reliable matching between ontologies. The evolution requires a fitness function in order to gauge the goodness of the intermediate solutions to the ontology matching problem.

A fitness function should utilize the lexical and structural similarity metrics to estimate the fineness of an alignment. The version of SANOM participated this year uses both lexical and structural similarity metrics, which are described in the following.

1.1 Lexical Similarity Metric

The cleaning of strings before the similarity computation is essential to increase the chance of mapping entities. SANOM uses the following pre-processing techniques to this end:

- **Tokenization.** It is quite common that the The terminology of concepts are constructed from a bag of words (BoW). The words are often concatenated by white space, the capital letter of first letters, and several punctuations such as " − " or "_". Therefore, they need to be broken into individual words and then the similarity is computed by comparing the bag of words together.

- **Stop word removal.** Stop words are the typical words with no particular meaning. The stop words should be detected by searching the tokens (identified after tokenization) in a table containing all possible stop words. The Glasgow stop word list is utilized in the current implementation [1] .
- **Stemming.** Two entities from the given ontologies might refer to a similar concept, but they are named differently due to various verb tense, plural/singular, and so forth. Therefore, one needs to recover the normal words so that the similar concepts will have higher similarity. The Porter stemming method is used for this matter [4].

After the pre-processing step, the strings of two concepts can be given to a similarity metric in order to calibrate the degree of similarity between concepts. The base similarity metric computes the sameness of tokens obtained from each entity. The current version of SANOM takes advantage of two similarity metrics and take their maximum as the final similarity of two given tokens. One of this similarity metric is for sole comparison of stirngs, and the other one is to guage the linguistic relation of two given names. These similarity metrics are:

- **Jaro-Winkler metric.** The combination of TF-IDF and Jaro-Winkler is popular and has been sucessful in ontology alignment as well. Similarly, SANOM uses Jaro-Winkler with the threshold 0.9 as one of the base similarity metrics.
- **WordNet-based metric.** The linguistic heterogeneity is also rampant in various domains. Therefore, the existence of a similarity metric to measure the lingual closeness of two entities is absolutely essential. In this study, the relatedness of two given tokens are computed by the Wu and Palmer measure [5] and is used as a base similarity metric with the threshold 0.95.

1.2 Structural Similarity Metric

The preceding string similarity metric gives a high score to the entities which have lexical or linguistic proximity. Another similarity of two entities could be derived from their positions in the given ontologies.

We consider two structural similarity measures for the current implementation of SANOM:

- The first structural similarity is gauged by the subsumption relation of classes. If there are two classes c_1 and c_2 whose superclasses are s_1 and s_2 from two given ontologies O_1 and O_2, then the matching of classes s_1 and s_2 would increase the similarity of c_1 and c_2. Let s be a correspondence mapping s_1 to s_2, then the increased similarity of c_1 and c_2 is gauged by

$$f_{structural}(c_1, c_2) = f(s). \tag{1}$$

---

[1] http://ir.dcs.gla.ac.uk/resources/linguistic utils/stop words

- Another structural similarity is derived from the properties of the given ontologies. The alignment of two properties would tell us the fact that their corresponding domain and/or ranges are also identical. Similarly, if two properties have the analogous domain and/or range, then it is likely that they are similar as well.

  The names of properties and even their corresponding core concepts are not a reliable meter based on which they are declared a correspondence. A recent study has shown that the mapping of properties solely based on their names would result in high false positive and false negative rates, e.g. there are properties with identical names which are not semantically related while there are semantically relevant properties with totally distinct names.

  The current implementation treats the object and data properties differently. For the object properties op_1 and op_2, their corresponding domains and ranges are computed as the concatenation of their set of ranges and domains, respectively. Then, the fitness of the names, domains, and ranges are computed by the Soft TF-IDF. The final mapping of two properties is the average of top two fitness scores obtained by the Soft TF-IDF. For the data properties, the fitness is computed as the similarity average of names and their corresponding domain.

  On the other flow of alignment, it is possible to derive if two classes are identical based on the properties. Let e_1 and e_2 be classes, op_1 and op_2 be the object properties, and R_1 and R_2 are the corresponding ranges, then the correspondence $c = (e_1, e_2)$ is evaluated as

$$f_{structural}(c) = \frac{f_{string}(R_1, R_2) + f_{string}(op_1, op_2)}{2}. \tag{2}$$

2 Results

This section contains the results obtained by SANOM on the anatomy and conference track.

2.1 Anatomy Track

The anatomy track is one of the earliest benchmarks in the OAEI. The task is about aligning the Adult Mouse anatomy and a part of NCI thesaurus containing the anatomy of humans. Each of the ontologies has approximately 3,000 classes, which are designed carefully and are annotated in technical terms.

The best performing systems in this track use a biomedical background knowledge. Thus, their results are not comparable with SANOM which does not use any particular background knowledge. Among other systems, LogMap [2] is best one with no use of a background knowledge.

Table 1 tabulates the precision, recall, and F-measure of SANOM and LogMap on the anatomy track. According to this table, the recall of SANOM is slightly higher than LogMap which means that it could identify more correspondences than LogMap. However, the precision of LogMap is better than SANOM with the margin of three percent. The overall performance of SANOM is quite close to LogMap since their F-measure has only 1% difference.

System	Precision	F-measure	Recall
LogMap	0.918	0.88	0.846
SANOM	0.888	0.87	0.853

Table 1: The precision, recall, and F-measure of SANOM and LogMap on the OAEI anatomy track.

	SANOM			AML			LogMap		
	P	F	R	P	F	R	P	F	R
cmt-conference	0.61	0.74	0.93	0.67	0.59	0.53	0.73	0.62	0.53
cmt-confOf	0.80	0.62	0.50	0.90	0.69	0.56	0.83	0.45	0.31
cmt-edas	0.63	0.69	0.77	0.90	0.78	0.69	0.89	0.73	0.62
cmt-ekaw	0.54	0.58	0.64	0.75	0.63	0.55	0.75	0.63	0.55
cmt-iasted	0.67	0.80	1.00	0.80	0.89	1.00	0.80	0.89	1.00
cmt-sigkdd	0.85	0.88	0.92	0.92	0.92	0.92	1.00	0.91	0.83
conference-confOf	0.79	0.76	0.73	0.87	0.87	0.87	0.85	0.79	0.73
conference-edas	0.67	0.74	0.82	0.73	0.69	0.65	0.85	0.73	0.65
conference-ekaw	0.66	0.70	0.76	0.78	0.75	0.72	0.63	0.55	0.48
conference-iasted	0.88	0.64	0.50	0.83	0.50	0.36	0.88	0.64	0.50
conference-sigkdd	0.75	0.77	0.80	0.85	0.79	0.73	0.85	0.79	0.73
confOf-edas	0.82	0.78	0.74	0.92	0.71	0.58	0.77	0.63	0.53
confOf-ekaw	0.81	0.83	0.85	0.94	0.86	0.80	0.93	0.80	0.70
confOf-iasted	0.71	0.63	0.56	0.80	0.57	0.44	1.00	0.62	0.44
confOf-sigkdd	0.83	0.77	0.71	1.00	0.92	0.86	1.00	0.83	0.71
edas-ekaw	0.71	0.72	0.74	0.79	0.59	0.48	0.75	0.62	0.52
edas-iasted	0.69	0.56	0.47	0.82	0.60	0.47	0.88	0.52	0.37
edas-sigkdd	0.80	0.64	0.53	1.00	0.80	0.67	0.88	0.61	0.47
ekaw-iasted	0.70	0.70	0.70	0.88	0.78	0.70	0.75	0.67	0.60
ekaw-sigkdd	0.89	0.80	0.73	0.80	0.76	0.73	0.86	0.67	0.55
iasted-sigkdd	0.70	0.80	0.93	0.81	0.84	0.87	0.71	0.69	0.67
Average	0.74	0.72	0.73	0.84	0.74	0.67	0.84	0.68	0.59

Table 2: The precision, recall, and F-measure of SANOM, AML, and LogMap on various datasets on the conference track

2.2 Conference Track

The conference comprises the pairwise alignment of seven ontologies. Table 2 displays the precision, recall, and F-measure of SANOM, LogMap, and AML [1] on the conference track. AML and LogMap are the top two systems in terms of precision and recall.

According to Table 2, the recall of SANOM is superior to both LogMap and AML. SANOM's average recall is 7% and 14% more than those of AML and LogMap, respectively, but its precision is 10% less than both of the systems. Overall, the performance of SANOM is quite competitive with the top performing systems in the conference track.

3 Conclusion

SANOM only participated in the OAEI 2018 anatomy and conference track. For the next year, we have aims to participate in more tracks so that the performance of SANOM can be compared with that of the state-of-the-art systems in other tracks as well. Another avenue to improve the system is to equip it with a proper biomedical background knowledge since most of the OAEI tracks are from this domain.

References

1. Daniel Faria, Booma Sowkarthiga Balasubramani, Vivek R Shivaprabhu, Isabela Mott, Catia Pesquita, Francisco M Couto, and Isabel F Cruz. Results of aml in oaei 2017. In *OM-2017: Proceedings of the Twelfth International Workshop on Ontology Matching*, page 122, 2017.
2. Ernesto Jiménez-Ruiz and Bernardo Cuenca Grau. Logmap: Logic-based and scalable ontology matching. In *International Semantic Web Conference*, pages 273–288. Springer, 2011.
3. Majid Mohammadi, Wout Hofman, and Yaohua Tan. Simulated annealing-based ontology matching. 2018.
4. Martin F Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.
5. Zhibiao Wu and Martha Palmer. Verbs semantics and lexical selection. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, pages 133–138. Association for Computational Linguistics, 1994.

XMap : Results for OAEI 2018

Warith Eddine DJEDDI[a, b], Sadok BEN YAHIA[b] and Mohamed Tarek KHADIR[a]

[a]LabGED, Computer Science Department, University Badji Mokhtar, Annaba, Algeria
[b]Faculty of Sciences of Tunis, University of Tunis El-Manar, LIPAH-LR 11ES14, 2092, Tunisia
{djeddi,khadir}@labged.net
sadok.benyahia@fst.rnu.tn

Abstract. We describe in this paper the XMap system and the results achieved during the 2018 edition of the Ontology Alignment Evaluation Initiative. XMap aims to tackle the issue of matching large scale ontologies by involving particular parallel matching on multiple cores or machines. Our strategies aim to provide a set of requirements that foster the using of a domain-specific thesaurus for the alignment of specialized ontologies.

1 Presentation of the system

The eXtended Mapping (XMap) algorithm relies on the context notion to deal with lexical ambiguity as well as a parallel comparison between concepts to efficiently handle the matching of large ontologies. Our approach to matching ontologies employs different components and steps in the ontology alignment process such as preprocessing, matching, filtering and combining matching results, and oracle validation of mapping suggestions. The contributions are the following:

- Defining a semantic similarity measure using UMLS[1] [1] and WordNet [2] to provide a synonymy degree between two entities from different ontologies, by exploring both of their lexical and structural contexts. In *XMap*, the measurement of lexical similarity in ontology matching is performed using a synset, defined in WordNet and UMLS. In our approach, the similarity between two entities of different ontologies is evaluated not only by investigating the semantics of the entity names, but also taking into account the context, through which the effective meaning is described. It is worth mentioning that the context is the set of information (partly) characterizing the situation of some entities [3]. The context notion is not universal but it is relative to some situations, tasks or applications [4, 5];
- Limiting the number of mapping suggestions to be validated by an oracle. Indeed, our approach employs a double threshold to produce matching candidates and use a small set of constraints [6, 7] (e.g., consistency, locality, and conservativity or quality checks), acting as a filter to select the final alignments. The first threshold is used at the interactive selection algorithm, which will ask the oracle for feedback about mappings when they are below a given similarity threshold, until a given number of negative answers is reached. The second threshold is used at the final

---

[1] `http://www.nlm.nih.gov/research/umls/`

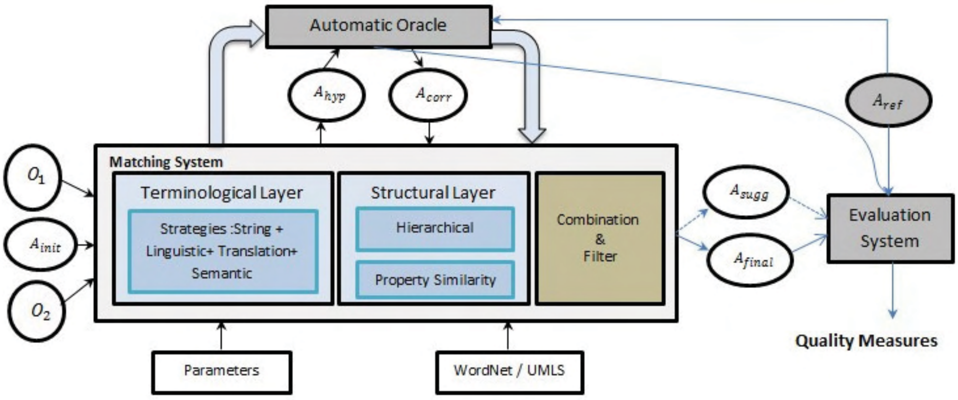


Fig. 1. The different steps for scoring a multiple network alignment.

stage to filter out the set of correspondences having a similarity value below a given threshold. This strategy skips over the problem of the growing size and the complexity of the user participation in the process alignment of large ontologies.
- Applying repair techniques from Applying Logical Constraints on Matching Ontologies (ALCOMO) [8] to make reference alignments coherent, by removing less unsatisfiable classes (discovering disjointness relationships) without having an impact on the F-measure score. Our strategy in the repair mode takes into account the confidence values during the selection of mappings to be removed in order to improve the quality of the repaired alignments in terms of computation time and mapping coherence.
- Finally, is the ability of XMap to deal with large scale ontology matching, by producing good experimental results in terms of quality of the alignments, time performance and scalability.

2 State, purpose, general statement

Our prototype leans on the architecture of a sequential/parallel composition. XMap uses various similarity measures of different categories such as string, linguistic, and structural based similarity measures, each contributing to some extent to the alignment results. At a glance, the mapping process of XMap is depicted in Figure 1. *XMap* receives as an input two source ontologies. The mappings discovered by the terminological level matcher are transferred to the structural level matcher in order to find new correspondences by analyzing the context of the entities in the taxonomy of ontologies. Afterwards, the combined result of the two basic matchers are aggregated by a weighted sum aggregation operator. For the final alignment method, the system uses the threshold method. Moreover, we manually define the filters threshold value to produce the final mappings. A fast repair method is applied so as to detect and remove the inconsistent ones.

3 Results

In this section, we present the evaluation results obtained by running XMap under the SEALS client with *Anatomy*, *Conference*, *Multifarm*, *Interactive matching evaluation*, *Large Biomedical Ontologies*, *Disease and Phenotype* and *Biodiversity and Ecology* tracks.

Anatomy The Anatomy track consists of finding an alignment between the Adult Mouse Anatomy (2744 classes) and a part of the NCI Thesaurus (3304 classes) describing the human anatomy. XMap achieves a good F-Measure value of ≈89% in a reasonable amount of time (37 sec.) (cf., Table 1).

Table 1. Results for Anatomy track.

System	Precision	F-Measure	Recall	Time(s)
XMap	0.929	0.896	0.865	37
StringEquiv	0.997	0.766	0.622	946

Conference The Conference track uses a collection of 16 ontologies from the domain of academic conferences. Most ontologies were equipped with OWL-DL axioms of various types; this opens a useful way to test our semantic matchers. For each reference alignment, three evaluation modalities are applied : a) crisp reference alignments, b) the uncertain version of the reference alignment, c) logical reasoning.

Table 2. Results based on the crisp reference alignments.

	Precision	F-Measure 1	Recall
	Original reference alignment (ra1)		
ra1-M1	0.81	0.70	0.61
ra1-M2	0.69	0.31	0.20
ra1-M3	0.81	0.65	0.54
	Entailed reference alignment (ra2)		
ra2-M1	0.79	0.65	0.55
ra2-M2	0.77	0.34	0.22
ra2-M3	0.77	0.61	0.5
	Violation reference alignment (rar2)		
rar2-M1	0.78	0.66	0.57
rar2-M2	0.77	0.34	0.22
rar2-M3	0.76	0.62	0.52

As depicted in Table 2 and 3, XMap produces fairly consistent alignments when matching the conference ontologies. Finally, XMap generated only one incoherent alignment for the evaluation based on logical reasoning.

Table 3. Results based on the uncertain version of the reference alignment.

Precision	F-Measure 1	Recall
Uncertain reference alignments (Sharp)		
0.81	0.65	0.54
Uncertain reference alignments (Discrete)		
0.66	0.74	0.83
Uncertain reference alignments (Continuous)		
0.74	0.70	0.66

Multifarm This track is based on the translation of the OntoFarm collection of ontologies into 9 different languages. XMap have low performance due to many internal exceptions. The results are showed in Table 4.

Table 4. Results for Multifarm track.

System	Different ontologies			Same ontologies		
	P	F	R	P	F	R
XMap	0.2	0.3	0.07	0.13	0.14	0.19

Interactive matching evaluation The goal of this evaluation is to imitate interactive alignment [9, 10], where a oracle user is involved to validate the correspondences found by the alignment approach by checking the reference alignment, and changing error values in order to assess their influence on the performance of alignment systems. For the 2018 edition, participating systems are evaluated on the *Conference* and *Anatomy* datasets using an oracle based on the reference alignment.

XMap uses various similarity measures to generate candidate mappings. It applies two thresholds to filter the candidate mappings: one for the mappings that are directly added to the final alignment and another for those that are presented to the user for validation. The latter threshold is selected to be high in order to minimize the number of requests and the rejected candidate mappings from the oracle; the requests are mainly about incorrect mappings. The mappings accepted by the user are moved to the final alignment. For the three years 2016, 2017 and 2018, XMap preserved roughly the same F-Measure value, and it benefits the least from the interaction with the oracle. Whereas, for the conference track, XMap has increases in precision, recall and F-measure. XMap's measures differ with less than 0.2% from the non-interactive runs, and performance does not change at all with the increasing error rates.

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). The results obtained by XMap (Evaluated without UMLS) are depicted by Table 5.

Table 5. Results for the Large BioMedical track.

Test set	Precision	Recall	F-Measure	Time(s)
Small FMA-NCI	0.977	0.783	0.869	7356
Whole FMA-NCI	0.877	0.741	0.803	66499
Small FMA-SNOMED	0.962	0.647	0.774	25544
Whole FMA- Large SNOMED	0.723	0.608	0.661	299027
Small SNOMED-NCI	0.835	0.588	0.69	123597
Whole SNOMED-NCI	0.64	0.582	0.61	426584

In general, we can conclude that XMap achieved a good precision/recall values. The high recall value can be explained by the fact that UMLS thesaurus contains definitions of highly technical medical terms.

Disease and Phenotype This track based on a real use case where it is required to find alignments between disease and phenotype ontologies. Specifically, the selected ontologies are the Human Phenotype Ontology (HPO), the Mammalian Phenotype Ontology (MP), the Human Disease Ontology (DOID), and the Orphanet and Rare Diseases Ontology (ORDO).

XMap achieved fair results according to the three evaluation (Silver standard, Manually generated mappings and Manual assessment of unique mappings).

Biodiversity and Ecology This track aims finding the alignments between the Environment Ontology (ENVO) and the Semantic Web for Earth and Environment Technology Ontology (SWEET), and between the Flora Phenotype Ontology (FLOPO) and the Plant Trait Ontology (PTO). The results are showed in Table .

Table 6. Results for the Biodiversity and Ecology track.

Test set	Precision	Recall	F-Measure	Time(s)
Small flopo-pto	0.987	0.761	0.619	153
Whole envo-sweet	0.868	0.785	0.716	547

4 General comments

4.1 Comments on the results

This is the 6th time that we participate in the OAEI campaign. The official results of OAEI 2018 show that XMap is competitive with other well-known ontology matching systems in all OAEI tracks.

4.2 Comments on the OAEI 2018 procedure

As a sixth participation, we found the OAEI procedure very convenient and the organizers very supportive. The OAEI test cases are various, and this leads to a comparison on different levels of difficulty, which is very interesting. We found that SEALS platform is a precious tool to compare the performance of our system with the others.

5 Conclusion

Generally, according to our results obtained during the compaing OAEI 2018, our system delivered good results comparatively to other well-known ontology matching systems. The used benchmark greatly helped to identify the power and weaknesses of the algorithm. used benchmark helped greatly identify the power and weaknesses of the algorithm. In addition, XMap showed the feasibility of our approach especially on large-scale biomedical ontologies which was a thriving challenge in ontology matching domain.

References

1. Olivier Bodenreider. The unified medical language system (UMLS): integrating biomedical terminology. *Nucleic Acids Research*, 32(Database-Issue):267–270, 2004.
2. Christiane D. Fellbaum. *WordNet – An Electronic Lexical Database*. MIT Press, 1998.
3. Anind K. Dey, Gregory D. Abowd, and Daniel Salber. A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. *Hum.-Comput. Interact.*, 16(2):97–166, December 2001.
4. Paul Dourish. Seeking a foundation for context-aware computing. *Human-Computer Interaction*, 16(2-4):229–241, 2001.
5. Matthew Chalmers. A historical view of context. *Computer Supported Cooperative Work*, 13(3):223–247, 2004.
6. Ernesto Jiménez-Ruiz, Bernardo Cuenca Grau, Ian Horrocks, and Rafael Berlanga Llavori. Logic-based assessment of the compatibility of UMLS ontology sources. *J. Biomedical Semantics*, 2(S-1):S2, 2011.
7. Elena Beisswanger and Udo Hahn. Towards valid and reusable reference alignments - ten basic quality checks for ontology alignments and their application to three different reference data sets. *J. Biomedical Semantics*, 3(S-1):S4, 2012.
8. Christian Meilicke. *Alignment incoherence in ontology matching*. PhD thesis, University of Mannheim, 2011.
9. Heiko Paulheim, Sven Hertling, and Dominique Ritzei. Towards evaluating interactive ontology matching tools. In *The Semantic Web: Semantics and Big Data, 10th International Conference, ESWC 2013, Montpellier, France, May 26-30, 2013. Proceedings*, pages 31–45, 2013.
10. Zlatan Dragisic, Valentina Ivanova, Patrick Lambrix, Daniel Faria, Ernesto Jimenez-Ruiz, and Catia Pesquita. User validation in ontology alignment. In *Proceedings of the International Semantic Web Conference*, volume 9981 of *LNCS*, October 2016.

Exploiting BabelNet for generating subsumption

Mouna Kamel[1], Daniela Schmidt[2], Cassia Trojahn[1], and Renata Vieira[2]

[1] Institut de Recherche en Informatique de Toulouse, Toulouse, France
{mouna.kamel,cassia.trojahn}@irit.fr
[2] Pontificia Universidade Catolica do Rio Grande do Sul, Porto Alegre
daniela.schmidt@acad.pucrs.br, renata.vieira@pucrs.br

1 Introduction

Whereas the ontology matching field has developed fully in the last decades, most matching approaches are still limited to generating equivalences between entities of different ontologies. However, for many tasks, finding subsumption relations may be useful. Despite the variety of matching approaches in the literature, most of them rely on string-based techniques as an initial estimate of the likelihood that two elements refer to the same real world phenomenon, hence, the found correspondences represent equivalences with terms similarly written rather than subsumptions. This paper presents an approach relying on background knowledge from BabelNet (BN) [3] and on the notion of *context*. The latter has been exploited in different ways in ontology matching [2, 4]. They are used for disambiguating the senses that better express the meaning of ontology concepts when looking for subsumption relations between them in BN.

2 Proposed approach

The matching process is divided in two steps. The first step disambiguates the ontology concept, and the second looks for a subsumption relation between two concepts.

Concept disambiguation. It finds the semantically closer BN synset for a concept. We adopt the notion of *context* as a *bag of words*. For each ontology concept c, from the source s and target t ontologies, the context ctx_c is constructed from the available information about the concept (ID, labels, information on super and sub-concepts, etc.). The context of BN synsets ctx_{bn} is constructed from their sense and main glosses terms. We adapt the word sense disambiguation method of Lesk [1], which relies on the calculation of the word overlap between the sense definitions of two or more target terms. Here, we overlap the context ctx_c and all ctx_{bn}, coming from the synsets retrieved when looking for c in BN. We retrieve the highest overlap. The overlap function is based on the edit distance similarity between words rather than on the exact match.

Subsumption detection. Given c_s and c_t concepts from the source and target ontologies, and their respectively retrieved synsets syn_s and syn_t obtained in the previous step, we look for a subsumption relation between c_s and c_t. For that purpose, we check if syn_t belongs to the set of hypernyms $Hyper(syn_s)$, where $Hyper(syn_s)$= $\bigcup_k Hyper^k(syn_s)$ and k is length of the path from syn_s to one of its hypernym synsets, based on a depth-first search strategy.

3 Experimentation

Material and methods. We used the set of 7 ontologies from the OAEI conference data set that are involved in the 21 available reference alignments. In our experiments, compounds with no entry in BN have been pre-processed by removing the modifiers (e.g. “Invited speaker” is a “Speaker”). We empirically selected k=2 for the path length and 0.8 as edit distance threshold. We used as reference the subsumptions inferred from the available equivalence reference alignments, using Hermit and the Alignment API 4.5. As many concepts do not have any super or sub concepts, we considered 2 settings: contexts as introduced above and the whole ontology as context for each concept. The best results, which are reported here, were obtained with the latter.

Results and discussion. Table 1 shows the results (measures were computed using the Alignment API). Overall, the best results are obtained when considering alignments close to those expected (extended and semantic measures) rather than exact ones. Looking at the results for each pair of ontologies, the best results where obtained for different pairs when using the different measures: *edas-ekaw* (classical), *confOf-edas* (extended) and *conference-sigkdd* (semantic). The overall low results are mainly due to two reasons: a high number of concepts can not be found in BN and using the modifier does help so much in this task; the construction of contexts suffers from the lack of annotations in the ontologies (as well many concepts do not have any super or sub concepts), and hence, contexts are not rich enough for disambiguating the synsets.

Table 1. Results for the 21 pairs (and those discarding empty alignments) and best pair results.

Average (21 pairs)						Best pair results					
Classical		Extended		Semantic		Classical		Extended		Semantic	
Prec	Rec	Prec	Rec	Prec	Rec	Prec	Rec	Prec	Rec	Prec	Rec
.06 (.23)	.02 (.07)	.14 (.16)	.05 (.06)	.02 (.02)	.22 (.22)	.22	.08	.50	.11	.14	.15

4 Conclusions

We presented an approach for generating subsumption correspondences relying on BabelNet. This task is still a gap in the field and the initial results presented here can be improved in different ways. We plan to improve the disambiguation strategy, exploiting word embeddings, to automatically enrich the ontology with annotations, to adopt a hybrid approach combining both lexical and background knowledge, to work on the confidence of the correspondences, and to look for other relations like meronymy.

References

1. M. Lesk. Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone. In *SIGDOC*, pages 24–26. ACM, 1986.
2. A. Maedche and S. Staab. Measuring Similarity between Ontologies. In *13th Conf. on Knowledge Engineering and Knowledge Management*, pages 251–263, 2002.
3. R. Navigli and S. P. Ponzetto. BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *AI*, 193:217–250, 2012.
4. F. C. Schadd and N. Roos. Coupling of Wordnet Entries for Ontology Mapping Using Virtual Documents. In *7th Workshop on Ontology Matching*, pages 25–36, 2012.

Medical Knowledge Graph Construction by Aligning Large Biomedical Datasets

Giorgos Stoilos, David Geleta,
Jetendr Shamdasani, and Mohammad Khodadadi

Babylon Health, London, SW3 3DD, UK
`firstname.lastname@babylonhealth.com`

1 Extended Abstract

Building large Knowledge Bases can be realised by aligning and integrating existing data sources. To support AI-based digital healthcare services within Babylon Health[1] significant effort to build a large medical KB was recently undertaken. To realise this goal a highly configurable and modular ontology integration pipeline has been created which works as follows: an initial ontology is used as a seed KB ($\mathcal{KB}_0$) and additional data sources are integrated into it creating new extended versions of $\mathcal{KB}_0$. The integration process is based on a *Matching* phase, an *Aggregation* phrase, and a final *PostProcessing* phase. In the *Matching* phase the following matchers can be used:

- An in-house LabelMatcher which is based along similar ideas as the label matcher in [1], i.e., label normalisation, inverted indexes, and more.
- The state-of-the-art systems AML [1] and LogMap [3] in both its versions LogMap$_o$[2] and LogMap$_c$[3].
- A UMLS-synonym and a UMLS-CUI based matcher, or mappings from 3rd parties like BioPortal, NHS, and more.

The mappings from the previous stage are *Aggregated* using a weighted average and a threshold is applied. Finally, post-processing performs the following:

- Mappings of higher-multiplicity (i.e., mapping multiple classes to the same one) are separated from the rest. The former are handled by *multiplicity-disambiguation* techniques which reduce them to 1-to-1 or 1-to-m mappings.
- All mappings go through existing [2] and novel [4] conservativity-based mapping repair methods in order to avoid altering the structure of the seed KB.

Significant efforts were spent to determine which matching algorithm to use in the *Matching* phase. The Large BioMedTrack datasets were considered for evaluating the methods, however, surprisingly enough these datasets are much older, smaller and with somewhat different content compared to the recent releases of

---

[1] `https://www.babylonhealth.com/`

[2] `https://github.com/ernestojimenezruiz/logmap-matcher`

[3] `https://github.com/asolimando/logmap-conservativity/`

Table 1. Evaluation results on aligning official releases of SNOMED and NCI

	precision	recall	f-Value	Time(sec)	♯mapppings
LabelMatcher	0.356	0.77	0.49	13	28457
LogMap	0.372	0.78	0.50	2 850	27342
AML	0.410	0.50	0.45	596	15861

Table 2. Statistics about the KB after each integration/enrichment iteration.

	SNOMED	+NCI	+CHV	+FMA
Classes	340 995	429 241	429 241	524 837
Properties	93	124	124	219
$\|A \sqsubseteq B\|$	511 656	617 542	617 542	713 313
$\|\langle A\ p\ \mathsf{iri} \cup \mathsf{Lit}\rangle\|$	1 069 562	1 611 543	1 708 616	2 173 649

SNOMED, NCI, and FMA that are considered in Babylon. For example, NCI in BioTrack is almost half the size of the NCI December 2017 release (the former contains 96K axioms whereas the latter 185K), FMA is almost 1/4 and SNOMED almost 1/3 of their recent releases. In addition, synonym labels of classes seem to be completely missing from all ontologies. For those reasons the reference set between SNOMED and NCI in the BioTrack was refactored to point to codes in the official releases and then a precision/recall evaluation of our LabelMatcher, AML, LogMap, and XMap was conduced using the official releases (see Table 1); XMap did not manage to terminate.

As can be seen, although in theory simple, LabelMatcher provides comparable precision/recall and is orders of magnitude faster; the very low precision is because of the extra mappings found in the larger ontology versions which are confused as false positives. Given the scalability results and adequate precision/recall, we used our LabelMatcher in the pipeline to integrate the latest versions of NCI, CHV, and FMA on top of SNOMED (indeed this process could not be completed using AML or LogMap$_o$). Statistics about the KBs that we created after each integration are depicted in Table 2; moreover, no conservativity violations could be detected due to our post-processing.

We have also compared our post-processing approach against mapping repairing implemented in AML, LogMap$_c$ and LogMap$_o$. In cases that these systems don't terminate we used smaller versions of our (test) ontologies. In all cases a large number of conservativity violations could be identified (in contrast to none detectable after running our approach); detailed results can be found in [4].

References

1. Faria, D., Pesquita, C., Santos, E., Palmonari, M., Cruz, I.F., Couto, F.M.: The agreementmakerlight ontology matching system. In: Proc. of OTM (2013)
2. Jiménez-Ruiz, E., Grau, B.C., Horrocks, I., Llavori, R.B.: Ontology integration using mappings: Towards getting the right logical consequences. In: Proc. of ESWC (2009)
3. Jiménez-Ruiz, E., Grau, B.C., Zhou, Y.: Logmap 2.0: towards logic-based, scalable and interactive ontology matching. In: Proc. of SWAT4(HC)LS. pp. 45–46 (2011)
4. Stoilos, G., Geleta, D., Shamdasani, J., Khodadadi, M.: A novel approach and practical algorithms for ontology integration. In: Proceedings of ISWC (2018)

Partitioning and Matching Tuning of Large Biomedical Ontologies

Amir Laadhar[1], Faiza Ghozzi[2], Ryutaro Ichise[3], Imen Megdiche[1], Franck Ravat[1], and Olivier Teste[1]

[1] Toulouse University, IRIT (CNRS/UMR 5505), Toulouse, France
{firstname.lastname}@irit.fr
[2] University of Sfax, MIRACL, Sfax, Tunisia
faiza.ghozzi@isims.usf.tn
[3] National Institute of Informatics, Tokyo, Japan
ichise@nii.ac.jp

1 Introduction

Large biomedical ontologies such as SNOMED CT, NCI, and FMA are extensively employed in the biomedical domain. These complex ontologies are based on diverse modelling views and vocabularies. We define an approach that breaks up a large ontology alignment problem into a set of smaller matching tasks. We coupled this approach with an automated tuning process, which generates the adequate thresholds of the available similarity measure for any biomedical matching task. Experiments demonstrate that the coupling between ontology partitioning and threshold tuning outperforms the existing approaches.

2 Partitioning and Matching Tuning of Biomedical Ontologies

2.1 Architecture overview

In figure 1, we depict the different stages for ontologies partitioning and threshold tuning. These stages are detailed in the following sections.

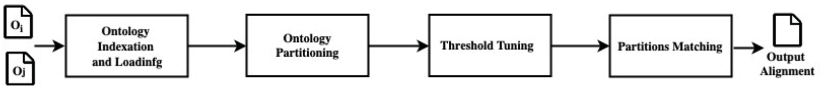


Fig. 1. Architecture Overview

2.2 Ontologies Partitioning

We employ the hierarchical agglomerative clustering technique to divide an ontology into a set of partitions. This method is based on the equation 1 to compute the structural similarity between the entities of the input ontologies. This equation is inspired by Wu and Palmer [4] similarity measure. The partitioning of every ontology results in a dendrogram. We cut each dendrogram automatically in order to result in a set of partitions. We examine the output of all the possible cuts until finding the first cut which do not result in any isolated partitions. Isolated partitions are partitions containing only one entity. We identify the similar partition-pairs through the set of exact matchings between the input ontologies.

$$StrcSim(e_{i,m}, e_{i,n}) = \frac{Dist(r_i, lca) \times 2}{Dist(e_{i,m}, lca) + Dist(e_{i,n}, lca) + Dist(r_i, lca) \times 2} \quad (1)$$

2.3 Threshold tuning

The available external knowledge sources represent mediator biomedical ontologies between the two input ontologies. We cross-search the input ontologies and the mediating ontology in order to find synthetic reference alignments. We compute the similarity score *Sim* between all the annotations of the generated alignments. These similarity scores are represented by: $simScore = \left\{ sim_1,\ldots, sim_n \right\}$. The threshold T_h value is deducted from *simScore* using the Equation 2:

$$T_h = \frac{\sum_{sim_1}^{sim_n} sim_i}{|simScore|} \tag{2}$$

3 Experiments

In Table 1, we compare our proposed partitioning approach to the currently available partitioning strategies using two OAEI 2017 biomedical data sets: the Anatomy task and the LargeBio small segments tasks.

Table 1. Anatomy track partitioning results

	Precision	F-Measure	Recall	Number of partitions
Proposed approach	0.945	0.883	0.829	57/57
SeeCOnt [3]	0.951	0.863	0.789	ND
Falcon [2]	0.964	0.730	0.591	139/119
Alsayed et al. [1]	0.975	0.753	0.613	84/80

We employed UBERON as an external biomedical knowledge for deriving synthetic reference alignments. We use ISUB similarity measure to compute the similarity scores between the derived mappings. In Table 2, we illustrate the accuracy of the partitioning approach with the deduced thresholds.

Table 2. Accuracy and derived thresholds for Anatomy and LargeBio tracks

	Precision	F-Measure	Recall	Derived Threshold
Anatomy	0.945	0.883	0.829	0.91
FMA-NCI	0.957	0.870	0.789	0.69
FMA-SNOMED	0.860	0.674	0.554	0.75
SNOMED-NCI	0.911	0.697	0.564	0.85

4 Conclusion and Future Work

As future work, we intend to automate all the matching tuning process while focusing on different type of heterogeneity applied over the partitions-pairs.

References

1. Algergawy, Alsayed, Sabine Massmann, and Erhard Rahm. ”A clustering-based approach for large-scale ontology matching.” East European Conference on Advances in Databases and Information Systems. Springer, Berlin, Heidelberg, (2011).
2. Hu, Wei, Yuzhong Qu, and Gong Cheng. ”Matching large ontologies: A divide-and-conquer approach.” Data Knowledge Engineering 67.1, (2008).
3. Algergawy, Alsayed, Samira Babalou, Mohammad J. Kargar, and S. Hashem Davarpanah. ”Seecont: A new seeding-based clustering approach for ontology matching.” In East European Conference on Advances in Databases and Information Systems, Springer (2015).
4. Wu, Zhibiao, and Martha Palmer. ”Verbs semantics and lexical selection.” In Proceedings of the 32nd annual meeting on Association for Computational Linguistics, (1994).

Complex matching for multiple ontologies: an exploratory study

Madalena Pavão and Cátia Pesquita

LASIGE, Faculdade de Ciências da Universidade de Lisboa, Portugal
`mpavao@lasige.di.fc.ul.pt, clpesquita@ciencias.ulisboa.pt`

1 Introduction

Complex matching, i.e., finding correspondences that go beyond equivalence and are able to capture more complex relationships between entities or sets of entities, is a recognized challenge (for a more in-depth overview see [1]). However, in multi-domain areas, there is a need to perform 'holistic' matching - link multiple ontologies to address different perspectives of the underlying data, while maintaining the inherently distributed paradigm championed by the Semantic Web. This need motivates 'compound mappings' involving more than two ontologies. A specific case is the ternary compound mapping [2], for example, the HP class 'broad forehead' is equivalent to an axiom obtained by relating PATO ('increased width') and UBERON ('forehead') classes, via an intersection. In this study, we explore compound matching involving multiple concepts from multiple ontologies.

2 Methods and Results

A compound mapping is a tuple $<C_s$,[C_{t0},...,C_{tn}],[P_{t0},...,P_{tn}],M$>$, where C_s is a class from a source ontology, [C_{t0},...,C_{tn}] and [P_{t0},...,P_{tn}] are a set of target classes extracted from multiple target ontologies and the set of properties that stand between them, while M is a mapping relation established between the source class and the expression composed by the set of target classes and properties. For our purposes, we are restricting our approach to finding mappings where M is an equivalence, and we are simplifying our goal to just finding the set of target classes. Our compound matching algorithm is based on finding partial lexical matches between word sequences in the source class labels and full labels of target classes. The algorithms for compound alignment were developed within the AgreementMakerLight (AML) system [3] and use hash map-based data structures to improve scalability.

We evaluated our algorithm using four test cases: (1) HP as source, UBERON and PATO as targets; (2) HP as source, UBERON, PATO and GO as targets; (3) MP as source, UBERON and PATO as targets; (4) MP as source, UBERON, PATO and GO as targets[1]. The reference alignments for each task were generated

---

[1] HP:Human Phenotype Ontology; PATO: Phenotypic Quality Ontology; UBERON: Uber anatomy ontology; GO:Gene Ontology

by extracting all the Equivalent Classes Axioms of MP and HP OWL files with OWL API. For each ontology we created two references: (1) UB-PT: containing mappings that employ classes from the UBERON and/or PATO ontologies; (2) UB-PT-GO: containing all mappings that employ the UBERON and/or PATO and/or GO ontologies. Note that these are just partial alignments, since they only cover 39%of the classes in HP and 28.7% in MP.

In our evaluation (shown in Table 1) we have considered partial mappings as positive[2]. The MP tasks had a better performance than HP, and for MP there was a marked difference between the UB-PT and UB-PT-GO, with the former having an improvement of over 30% in f-measure.

Table 1. Performance metrics for partial mappings

	Precision	**Recall**	**F-measure**
MP-UB-PT	75.4%	98.8%	85.6%
MP-UB-PT-GO	46.4%	61.1%	52.8%
HP-UB-PT	26.0%	43.4%	32.5%
HP-UB-PT-GO	26.1%	44.9%	33.0%

3 Conclusions

This exploratory study highlights the difficulties in performing holistic matching, which range from scalability issues to the inherent increased complexity of the task. Recent evaluation of complex matching approaches revealed that all techniques produced f-measures below 20% [1]. Furthermore, building reference alignments for compound matching is a standing challenge, and the reference alignments we built can only be considered partial references. We envision several future work endeavors in this area, ranging from more complex lexical approaches to approaches richer in semantics that are able to capture property restrictions to reproduce the full equivalence axiom.

Acknowledgments This work has been supported by the Fundação da Ciência e Tecnologia through funding of LaSIGE Research Unit, ref.UID/CEC/00408/2013 and by the project SMiLaX (PTDC/EEI-ESS/4633/2014).

References

1. Thiéblin, É., Haemmerlé, O., Hernandez, N., Trojahn, C.: Task-oriented complex ontology alignment: Two alignment evaluation sets. In: European Semantic Web Conference, Springer (2018) 655–670
2. Oliveira, D., Pesquita, C.: Improving the interoperability of biomedical ontologies with compound alignments. Journal of biomedical semantics **9**(1) (2018) 1
3. 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, Berlin, Heidelberg, Springer (2013) 527–541

---

[2] Mappings that have at least one class in common with the classes in the reference

MCHA SPAIDA: A Cooperative Query Editor with Anonymous Helpers using Ontology Mappings

Takuya Adachi[1] and Naoki Fukuta[2]

[1] Department of Informatics, Graduate School of Integrated Science and Technology, Shizuoka University
[2] College of Informatics, Academic Institute, Shizuoka University
`{adachi.takuya.17@,fukuta@inf.}shizuoka.ac.jp`

Abstract. In this paper, we propose our prototype system, named MCHA SPAIDA, that allows us for cooperatively editing SPARQL queries by the help of anonymous helper users without revealing the detailed meaning of the query. This system dynamically generates ontology mappings to translate a modified query by using a query and some data on an endpoint. Ontology matching approaches have been applied to effectively anonymize the query to be cooperatively edited by other users without loss of semantic relations among data and vocabularies. To make ease of cooperative tracing and profiling of a query, our method will not directly modify the queries and given ontology mappings. Rather, our method tries to add a small amount of supplemental ontology mappings to effectively anonymize the meaning of original query.

Keywords: SPARQL, ontology mapping, cooperative query editing, privacy protection, query anonymization

1 MCHA SPAIDA

To overcome the issues, we are implementing a system named MCHA SPAIDA[3], which is an extended version of our previously implemented system SPAIDA for utilizing ontology mappings on SPARQL queries[1–3] which also includes anonymous helper mechanism MCHA for cooperatively editing and sophisticating queries. Our system includes an "on-the-fly" ontology and instance matcher to evaluate the used ontology mappings and instance mappings, suggests which mappings will be used in the query. Furthermore, the on-the-fly matcher can also be used to interactively add one-time mappings that could produce more complete answer in the results of the specified query [1].

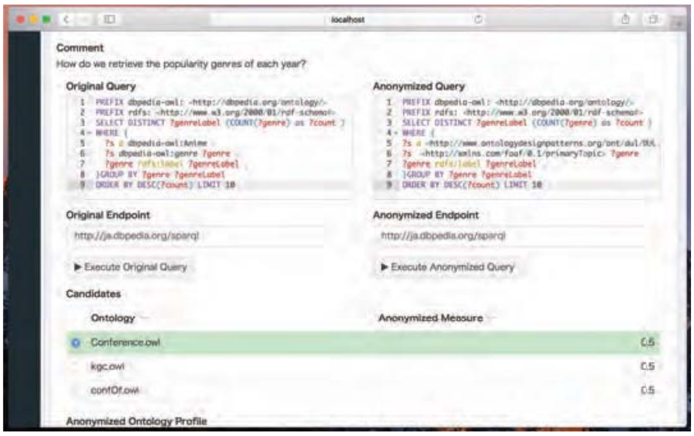
We are implementing a prototype system as a web application with SPARQL query editors and anonymous helpers. In order to utilize alignments as mapping data by using ontology matching tools (e.g., Alignment API [4]) and instance

---

[3] A demo is available at http://whitebear.cs.inf.shizuoka.ac.jp/spaida-demo/

matching tools (e.g., ScLink [5]), SPAIDA prepares mapping repositories to the outside.

Figure 1 shows an overview of our implementing system.



Comment
How do we retrieve the popularity genres of each year?

Original Query

```
PREFIX dbpedia-owl: <http://dbpedia.org/ontology/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
SELECT DISTINCT ?genreLabel (COUNT(?genre) as ?count )
WHERE {
  ?s a dbpedia-owl:Anime .
  ?s dbpedia-owl:genre ?genre .
  ?genre rdfs:label ?genreLabel .
}GROUP BY ?genre ?genreLabel
ORDER BY DESC(?count) LIMIT 10
```

Anonymized Query

```
PREFIX dbpedia-owl: <http://dbpedia.org/ontology/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
SELECT DISTINCT ?genreLabel (COUNT(?genre) as ?count )
WHERE {
?s a <http://www.ontologydesignpatterns.org/ont/dul/DUL.
?s  <http://xmlns.com/foaf/0.1/primaryTopic> ?genre
?genre rdfs:label ?genreLabel .
}GROUP BY ?genre ?genreLabel
ORDER BY DESC(?count) LIMIT 10
```

Original Endpoint
http://ja.dbpedia.org/sparql

Anonymized Endpoint
http://ja.dbpedia.org/sparql

Candidates

Ontology	Anonymized Measure
● Conference.owl	0.5
kgc.owl	0.5
confOf.owl	0.5

Fig. 1. An overview of our implementing system

2 Dynamic Mapping Generation

An application of the on-the-fly mapping generation mechanism is MCHA (Mapping-based Conversion for Human-based query writing Assistance), which rather convert a query to another query which targets to completely different things while it tries to keep their attributes in the sense of complexity and structure of the output. This allows anonymous cooperative helper editing of a query while mitigating the targets the original user is trying to access. In this mechanism, some users are asked to help editing and enhancing a query of mapping-based conversion of a query to a "semantically equivalent or very similar" query.

References

1. Adachi, T., Fukuta, N.: Toward Better Debugging Support on Extended SPARQL queries with On-the-fly Ontology Mapping Generation. In: Proc. of The 11th International Workshop on Ontology Matching (OM2016). pp. 239–240 (2016), (poster)
2. Adachi, T., Fukuta, N.: A Mapping-enhanced Linked Data Inspection and Querying Support System using Dynamic Ontology Matching. In: Proc. of 2nd International Workshop on Platforms and Applications for Social problem Solving and Collective Reasoning (PASSCR2017). pp. 1191–1194 (2017)
3. Adachi, T., Yamada, N., Fukuta, N.: Towards Better Query Coding Support Utilizing Ontology Mappings. In: Proc. of 1st International Workshop on Platforms and Applications for Social problem Solving and Collective Reasoning (PASSCR2016). pp. 96–99 (2016)
4. David, J., Euzenat, J., Scharffe, F., dos Santos, C.T.: The Alignment API 4.0. In: Semantic Web Journal 2(1), pp. 3–10. IOS Press (2011)
5. Nguyen, K., Ichise, R.: ScLink: Supervised Instance Matching System for Heterogeneous Repositories. In: Journal of Intelligent Information Systems, pp. 1–33. Springer US (2016). https://doi.org/10.1007/s10844-016-0426-3

Joint Handling of Semantic Knowledge Resources and their Alignments [⋆]

Bruno THIAO-LAYEL[1,3], Vianney JOUHET[1,2], and Gayo DIALLO[1]

[1] BPH Center, INSERM U1219, Team ERIAS, Univ. Bordeaux, France
[2] CHU de Bordeaux, Univ. Bordeaux, Bordeaux, France
[3] Synapse Medicine, 2 place de la bourse, Bordeaux, France
`first.last@u-bordeaux.fr`

1 Motivation

With the adoption of the Semantic Web vision , semantic knowledge resources (KR), which include taxonomies, structured vocabularies and ontologies, acting as pivotal resources, are nowadays commonly used. This is a direct consequence of the desire to attach formal semantic meaning to manipulated data. These KR, developed by different communities with various needs and purposes, are by nature heterogeneous. This heterogeneity leads to the development of systems for finding the correspondences between entities of different KR, called alignment [2]. In addition, KR increasingly involve large volumes of formalized knowledge, containing hundred of thousand entities, which raises the question of generating and validating alignments between large resources.

In order to help finding and reusing these more and more available KRs, a significant effort has been made for providing multi-knowledge resources repositories. This effort is particularly noticeable in the biomedical domain, where classifying existing objects is a secular tradition.

However, to the best of our knowledge, there is no currently available framework which offers the possibility to handle both multiple KRs together with their respective alignments, while keeping their native semantics and offering a support for a transparent visualization of these resources. In addition, with the development of the ontology matching domain, as different systems could be used to generate alignments and sometimes relying on user input, either for mappings validation purpose or initial alignment providing, it is a crucial issue to keep track of users and involved alignment methods or tools.

To fill this gap, we have designed the K-Ware framework [1] which complement existing multi-knowledge resources repositories. Its aim is not to provide a single access point for all available biomedical ontologies and alignments. Rather, it is a framework which could be embedded within projects that have an extensive use of multiple KRs and their respective alignments. In particular, enabling a support for multi point of view navigation and hierarchical visualization of any KR relevant for a dedicated purpose or suitable for a given project.

[⋆] This study is supported by the Drugs Systematized Assessment in real-liFe Environment (DrugSafe) platform, funded by the French Agency for Drugs Monitoring.

2 Joint handling of knowledge resources and alignments

2.1 Alignment management

In order to properly take into account alignments between different KRs, we have introduced additional properties to the definition of the notion of correspondence introduced in [2]. Therefore, we identify the following components for a correspondence : i) the two entities to be mapped, linked with a **MappingRelation**, in an directed order (*e1-mappingRelation-e2*); ii) its confidence value; iii) the mapping's author / **User**; iv) the mapping's method called **MappingMethod**, either if it is an alignment provided by an automated method **ComputerizedProcess** with an alignment **Tool** or a manual one; v) finally, a flag which indicates whether the considered mapping is valid (and validated by a **User**) for an **Alignment** between two **KRs**.

2.2 Handling structural relations within knowledge resources

KRs often exhibits a structural hierarchical organization : the *is-a* relationship translated into *rdfs:subClassOf* or the *part-whole* relationship for formal ontologies and the *narrower/broader* relationship for taxonomies, translated into *skos:narrower* and *skos:broader* respectively. We find similar notions when it comes to express inter relationships between entities in different KRs. For instance, the *skos:narrowMatch* is used to state a hierarchical mapping link between two conceptual resources in different concept schemes. Hierarchical aspect is the the main information to be kept when one wishes to integrate many different semantic resources.

To allow navigating easily KR represented in the OWL or SKOS languages, we distinguish three types of relations : **HierarchicalRelation**, **LiteralDefinition** and **MappingRelation**. In a given KR, a hierarchical relation could be *rdfs:subClassOf*, *skos:broader*, *part_of*, etc.). A literal definition helps rendering a human readable description of an entity (*rdfs:label*, *skos:prefLabel*, etc.). While a mapping relation (for instance *owl:equivalentTo*, *skos:exactMatch*) handle correspondences between entities.

An API has been implemented for the features defined models of K-Ware[4].

References

1. Thiao-Layel, B., Jouhet, V., Diallo, G. K-Ware: vers une gestion conjointe de ressources sémantiques et leurs alignements 6ièmes Journées Francophone sur les Ontologies, Oct 2016, Bordeaux, France
Hepp, M., de Bruijn, J.
2. Shvaiko, P., Euzenat, J.: Ontology Matching: State of the Art and Future Challenges. IEEE Transactions on Knowledge and Data Engineering, Volume: 25, Issue: 1, Jan. 2013, Page(s): 158 - 176

---

[4] http://k-wa.re