Ontology Matching

OM-2017

Proceedings of the ISWC Workshop

Introduction

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

The workshop has three goals:

- To bring together leaders from *academia*, *industry* and *user institutions* to assess how academic advances are addressing real-world requirements. The workshop strives to improve academic awareness of industrial and final user needs, and therefore, direct research towards those needs. Simultaneously, the workshop serves to inform industry and user representatives about existing research efforts that may meet their requirements. The workshop also investigated how the ontology matching technology is going to evolve.
- To conduct an extensive and rigorous evaluation of ontology matching and instance matching (link discovery) approaches through the OAEI (Ontology Alignment Evaluation Initiative) 2017 campaign[2]. Besides real-world specific matching tasks, such as the desease-phenotype track supported by the Pistoia Alliance, IBM Research sponsored the instance matching related tracks this year. Therefore, the ontology matching evaluation initiative itself provided a solid ground for discussion of how well the current approaches are meeting business needs.
- To examine new uses, similarities and differences from database schema matching, which has received decades of attention but is just beginning to transition to mainstream tools, or the emerging process matching task.

The program committee selected 5 submissions for oral presentation and 10 submissions for poster presentation. 21 matching systems participated in this year's OAEI campaign. Further information about the Ontology Matching workshop can be found at: `http://om2017.ontologymatching.org/`.

[1] `http://www.ontologymatching.org/`
[2] `http://oaei.ontologymatching.org/2017`

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 Informatica Trentina[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 2017

[3] http://www.taslab.eu
[4] http://www.openlivinglabs.eu
[5] http://www.infotn.it
[6] www.seals-project.eu
[7] https://project-hobbit.eu/challenges/om2017/
[8] http://www.pistoiaalliance.org/projects/ontologies-mapping/
[9] http://oaei.ontologymatching.org/2017/ibm_prize.html

Organization

Organizing Committee

Pavel Shvaiko, Informatica Trentina SpA, Italy
Jérôme Euzenat, INRIA & University Grenoble Alpes, France
Ernesto Jiménez-Ruiz, 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, LRIMM, France
Olivier Bodenreider, National Library of Medicine, USA
Marco Combetto, Informatica Trentina, Italy
Valerie Cross, Miami University, USA
Warith Eddine Djeddi, LIPAH & LABGED, Tunisia
Jérôme David, University Grenoble Alpes & INRIA, France
Gayo Diallo, University of Bordeaux, France
Zlatan Dragisic, Linköpings Universitet, Sweden
Alfio Ferrara, University of Milan, Italy
Wei Hu, Nanjing University, China
Antoine Isaac, Vrije Universiteit Amsterdam & Europeana, Netherlands
Valentina Ivanova, Linköpings Universitet, Sweden
Ryutaro Ichise, National Institute of Informatics, Japan
Daniel Faria, Instituto Gulbenkian de Ciência, Portugal
Patrick Lambrix, Linköpings Universitet, Sweden
Juanzi Li, Tsinghua University, China
Vincenzo Maltese, University of Trento, Italy
Fiona McNeill, University of Edinburgh, UK
Andriy Nikolov, Open University, UK
Axel Ngonga, University of Leipzig, Germany
Catia Pesquita, University of Lisbon, Portugal
Dominique Ritze, University of Mannheim, Germany
Umberto Straccia, ISTI-C.N.R., Italy
Ondřej Zamazal, Prague University of Economics, Czech Republic
Cássia Trojahn, IRIT, France
Ludger van Elst, DFKI, Germany

Table of Contents

Technical Papers

OAEI Papers

Posters

A High-Performance Approach to String Similarity using Most Frequent K Characters

Andre Valdestilhas, Tommaso Soru, and Axel-Cyrille Ngonga Ngomo

AKSW/DICE, University of Leipzig, Germany
{valdestilhas,tsoru,ngonga}@informatik.uni-leipzig.de

Abstract. The amount of available data has been growing significantly over the last decades. Thus, linking entries across heterogeneous data sources such as databases or knowledge bases, becomes an increasingly difficult problem, in particular w.r.t. the runtime of these tasks. Consequently, it is of utmost importance to provide time-efficient approaches for similarity joins in the Web of Data. While a number of scalable approaches have been developed for various measures, the Most Frequent k Characters (MFKC) measure has not been tackled in previous works. We hence present a sequence of filters that allow discarding comparisons when executing bounded similarity computations without losing recall. Therewith, we can reduce the runtime of bounded similarity computations by approximately 70%. Our experiments with a single-threaded, a parallel and a GPU implementation of our filters suggest that our approach scales well even when dealing with millions of potential comparisons.

Keywords: Similarity Search; Blocking; String Matching

1 Introduction

The problem of managing heterogeneity at both the semantic and syntactic levels among various information resources [12,10] is one of the most difficult problems on the information age. This is substantiated by most of the database research self-assessment reports, which acknowledge that the hard question of semantic heterogeneity, that is of handling variations in meaning or ambiguity in entity interpretation, remains open [10]. In knowledge bases, Ontology Matching (OM) solutions address the semantic heterogeneity problem in two steps: (1) matching entities to determine an alignment, i.e., a set of correspondences, and (2) interpreting an alignment according to application needs, such as data translation or query answering. Record Linkage (RL) and, more recently, Link Discovery[1] (LD) solutions on the other hand aim to determine pairs of entries that abide by a given relation R. In both cases, string similarities are used to compute

[1] The expression "link discovery" in this paper means the discovery of typed relations that link instances from knowledge bases on the Web of Data. We never use it in the sense of graph theory.

candidates for alignments. In addition to being central to RL and LD, these similarities also play a key role in several other tasks such as data translation, ontology merging and navigation on the Web of Data [10,2].

One of the core tasks when developing time-efficient RL and LD solutions hence lies in the development of time-efficient string similarities. In this paper, we study the MFKC similarity function [8] and present an approach for improving the performance of similarity joins. To this end, we develop a series of filters which guarantee that particular pairs of resources do not abide by their respective similarity threshold by virtue of their properties.

The contributions of this paper are as follows:

1. We present two nested filters, (1) First Frequency Filter and (2) Hash Intersection filter, that allow to discard candidates before calculating the actual similarity value, thus giving a considerable performance gain.
2. We present the *k similarity filter* that allows detecting whether two strings s and t are similar in a fewer number of steps.
3. We evaluate our approach with respect to its runtime and its scalability with several threshold settings and dataset sizes.
4. We present several parallel implementations of our approach and show that they work well on problems where $|D_s \times D_t| \geq 10^5$ pairs.

The rest of the paper is structured as follows: In Section 2 related work is presented, where we focus on approaches that aim to improve the time-efficiency of the link discovery task. In Section 3, we present our nested filters, followed by the Section 4 with the Correctness and Completeness. Section 5 with the evaluation. In Section 6, we conclude.

2 State of the Art and related work

Our approach can be considered an extension of the state-of-the-art algorithm introduced in [8], which describes a string-based distance function (SDF) based on string hashing [9,7]. The naive approach of MFKC [8] is a metric for string comparison built on a hash function, which gets a string and outputs the most frequent two characters with their frequencies. This algorithm was used for text mining operations. The approach can be divided into two parts: (1) The hashing function is applied to both input strings, where the output is a string that contains the two most frequent characters; the first and third elements keep the characters and second and fourth elements keep the frequency of these characters. (2) The hashes are compared, where will return a real number between 0 and lim. By default $lim = 10$, since the probability of having ten occurrences of the two most frequent characters in common between two strings is low. If the output of the function is 10, this case indicates that there is no common character and any value below 10 means there are some common characters shared among the strings.

Our work is similar to the one presented in [13], which features a parallel processing framework for string similarity using filters to avoid unnecessary comparisons. Among the several types of string similarity, emerging works have been

done for measures such as Levenshtein-distance [6], which is a string distance function that calculates the minimum number of edit operations (i.e., delete, insert or update) to transform the first into the second string. The Jaccard Index [5], also called Jaccard coefficient, works on the bitwise operators, where the strings are treated at bit level. REEDED [11] was the first approach for the time-efficient execution of weighted edit distances.

3 Approach

Let us call $NaiveMFKC$ the function which computes the MFKC algorithm as described in [8]. Such function works with three parameters, i.e. two strings s and t and an integer lim and returns the sum of frequencies, where $\mathrm{f}(c_i, s)$ is a function that returns the frequency of the character c_i in the string s and $s \supseteq \{c_1, ..., c_n\}$, i.e $\mathrm{f}(a, "andrea") = 2$, because the character a has been found twice and the hash functions $h(s)$ and $h(t)$ containing the characters and their frequencies. The output of function is always positive, as shown in Equation (1).

$$NaiveMFKC(s, t, lim) = lim - \sum_{c_i \in h(s) \cap h(t)}^{2} f(c_i, s) + f(c_i, t) \tag{1}$$

Our work aims to reduce the runtime of computation of the MFKC similarity function. Here, we use a sequence of filters, which allow discarding similarity computations and imply in a reduction of runtime. As input, the algorithm receives datasets D_s and D_t, an integer number representing the k most frequent characters and a threshold $\theta \in [0, 1]$. The similarity score of the pair of strings from the Cartesian product from D_s and D_t must have a score greater or equal the threshold θ to be considered a good pair, i.e. for a given threshold θ, if the similarity function has a pair of strings with similarity score less than the threshold, $\sigma(s, t) < \theta$, we can discard the computation of the MFKC score for this pair. Our final result is a set which contains the pairs having similarity score greater than or equal to the threshold, i.e. $\sigma(s, t) \geq \theta$.

Our work studies the following problem: Given a threshold $\theta \in [0, 1]$ and two sets of strings D_s and D_t, compute the set $M' = \{(s, t, \sigma(s, t)) \in D_s \times D_t \times \mathbb{R}^+ : \sigma(s, t) \geq \theta\}$. Two categories of approaches can be considered to improve the runtime of measures: Lossy approaches return a subset M'' of M' which can be calculated efficiently but for which there are no guarantees that $M'' = M'$. Lossless approaches, on the other hand, ensure that their result set M'' is exactly the same as M'. In this paper, we present a lossless approach that targets the MFKC algorithm. Equation (2) shows our definition for the string similarity function σ for the MFKC.

$$\sigma(s, t) = \frac{\sum_{c_i \in h(s,k) \cap h(t,k)} \mathrm{f}(c_i, s) + \mathrm{f}(c_i, t)}{|s| + |t|} \tag{2}$$

where s and t are strings, such that $s, t \in \sum^*$, $\mathrm{f}(c_i, s)$ is a function that returns the frequency of the character c_i in the string s, where $s \supseteq \{c_1, ..., c_n\}$, k represents the limitation of the elements that belongs to the hashes; set $h(s, k) \cap h(t, k)$

means the intersection between the keys of hashes $h(s,k)$ and $h(t,k)$ (i.e., the most frequent K characters). We expect two steps to obtain the similarity score:

1. Firstly, we transform the strings s and t in two hashes using Most Frequent Character Hashing [8], according to the following example with $k = 3$:
   $s = aabbbcc \rightarrow \mathrm{h}(s,k) = \{b = 3, a = 2, c = 2\}$
   $t = bbccddee \rightarrow \mathrm{h}(t,k) = \{b = 2, c = 2, d = 2\}$
2. We calculate the sum of the character frequencies of matching characters on the hashes $h(s,k)$ and $h(t,k)$, then, we normalize dividing by the sum of the length of $|s|$ and $|t|$ resulting in a similarity score from 0 to 1 according to the Equation (3) and the resulting score should be greater or equals the threshold θ.

$$\sigma(s,t,k,\theta) = \frac{\sum_{c_i \in \mathrm{h}(s,k) \cap \mathrm{h}(t,k)} \mathrm{f}(c_i,s) + \mathrm{f}(c_i,t)}{|s| + |t|} \geq \theta \tag{3}$$

3.1 Improving the Runtime

In this section, the runtime of MFKC defined in Equation (2) is improved using filters where $\mathcal{N}$ is the output of first frequency filter, $\mathcal{L}$ is the output of hash intersection filter and $\mathcal{A}$ represents the output of the k similarity filter.

First Frequency Filter As specified in the definition of MFKC [8] this filter assumes that the hashes are already sorted in an descending way according to the frequencies of characters, therefore the first element of each hash has the highest frequency.

Theorem 1. *Showing that:*

$$\sigma(s,t) = \frac{\sum_{c_i \in \mathrm{h}(s,k) \cap \mathrm{h}(t,k)} \mathrm{f}(c_i,s) + \mathrm{f}(c_i,t)}{|s| + |t|} \leq \frac{h_1(s,k)k + |t|}{|s| + |t|} \tag{4}$$

implies that $\sigma(s,t) < \theta$.

Proof (Theorem 1). Let the intersection between hashes $h(t,k)$ and $h(s,k)$ be a set of characters from c_1 to c_n, such that Equation (5):

$$h(t,k) \cap h(s,k) = \{c_1, ..., c_n\} \tag{5}$$

According to the definition of the frequencies $\mathrm{f}(c_i,t)$ we have Equation (6):

$$t \supseteq \{c_1, ..., c_1, ..., c_n, ...c_n\} \tag{6}$$

where each c_i appears $\mathrm{f}(c_i,t)$ times, therefore:

$$\mathrm{f}(c_1,t) + ... + \mathrm{f}(c_n,t) \leq |t| \tag{7}$$

Also, as $n \leq k$, because $t \supseteq \{c_1, ..., c_n\}$, and $\mathrm{f}(c_i,s) \leq h_1(s,k)\ \forall_{i=1,...,n}$, then:

$$\mathrm{f}(c_1,s) + ... + \mathrm{f}(c_n,s) \leq h_1(s,k) + ... + h_1(s,k) = n(k) \leq k(h_1(s,k)) \tag{8}$$

Therefore, from Equation (7) and Equation (8), we obtain the Equation (9):

$$\frac{\sum_{c_i \in \mathrm{h}(s,k) \cap \mathrm{h}(t,k)} \mathrm{f}(c_i, s) + \mathrm{f}(c_i, t)}{|s| + |t|} = \frac{\sum_{i=1}^{n} \mathrm{f}(c_i, t) + \sum_{i=1}^{n} \mathrm{f}(c_i, s)}{|s| + |t|} \leq \frac{h_1(s,k)k + |t|}{|s| + |t|} \tag{9}$$

Consequently, the rule which the filter relies on is the following.

$$\langle s, t \rangle \notin \mathcal{N} \Rightarrow \langle s, t \rangle \notin D_s \times D_t \wedge \frac{h_1(s,k)k + |t|}{|s| + |t|} \leq \theta \tag{10}$$

Hash Intersection Filter In this filter, we check if the intersection between two hashes is an empty set, then the MFKC, represented by σ, will return a similarity score of 0 and we can avoid the computation of similarity in this case. Consequently, the rule which the filter relies on is the following.

$$\langle s, t \rangle \in \mathcal{L} \Rightarrow \langle s, t \rangle \in D_s \times D_t \wedge |h(s) \cap h(t)| > 0 \tag{11}$$

we also can say that the Equation (12) represents a valid implication.

$$h(s) \cap h(t) = \emptyset \Rightarrow \sigma(s, t) = 0 \tag{12}$$

The Equation (12) means that if the intersection between $h(s, k)$ and $h(t, k)$ is a empty set, this implies that the similarity score will be 0. That means there is no character matching, then there is no need to compute the similarity for this pair of strings.

K Similarity filter For all the pairs left, the similarity score among them is calculated. After that, the third filter selects the pairs whose similarity score is greater or equal than a threshold θ.

$$\langle s, t \rangle \in \mathcal{A} \Leftrightarrow \langle s, t \rangle \in \mathcal{N} \wedge \sigma(s, t) \geq \theta \tag{13}$$

This filter provides a validation and we show that the score of previous k similarity is always lower than the next k, according to the Equation (14) and in some cases when the similarity score is been reached before compute all elements $\in \mathrm{h}(s, k) \cap \mathrm{h}(t, k)$, thus saving computation in these cases.

Here k is also used as a index of similarity function $\sigma_k(s, t)$ in order to get the similarity of all cases of k, from 1 to k, also to show the monotonicity.

Therefore we can say that the computation of similarity score occurs until $\sigma_k(s, t) \geq \theta$.

We will demonstrate that the similarity score of previous k similarity is always lower than the next k similarity, for all $k \in \mathbb{Z}^* : k \leq |s \cap t|$.

$$\sigma_{k+1}(s, t) \geq \sigma_k(s, t) \tag{14}$$

We rewrote the equation for the first iteration, according to Equation (15)

$$\sigma_k(s,t) = \frac{\sum_{c_i \in \mathrm{h}(s,k) \cap \mathrm{h}(t,k), i=1}^{k} \mathrm{f}(c_i, s) + \mathrm{f}(c_i, t)}{|s| + |t|} \tag{15}$$

Let $k \in \mathbb{Z}_+$ be given and suppose the equation Equation (14) is true for $k+1$.

$$\sum_{c_i \in \mathrm{h}(s,k) \cap \mathrm{h}(t,k)}^{k} \big[\mathrm{f}(c_i, s) + \mathrm{f}(c_i, t)\big] + \mathrm{f}(c_{k+1}, s) + \mathrm{f}(c_{k+1}, t) \geq \sum_{c_i \in \mathrm{h}(s,k) \cap \mathrm{h}(t,k)}^{k} \mathrm{f}(c_i, s) + \mathrm{f}(c_i, t) \tag{16}$$

Therefore, we can notice that the sum of frequencies will be always greater or equal 0, according to $\mathrm{f}(c_{k+1}, s) + \mathrm{f}(c_{k+1}, t) \geq 0$. Thus, Equation (14) holds true.

Filter sequence The sequence of the filters occurs basically in 4 steps, (1) We starting to make the Cartesian product with the pairs of strings from the datasets D_s and D_t, (2) Discarding pairs using the $First\ Frequency\ Filter(\mathcal{N})$, (3) Discarding pairs where there is no matching characters with the $Hash\ Intersection\ filter(\mathcal{L})$ and (4) With the Most Frequent Character Filter ($\mathcal{A}$) we will process only similarities greater or equal the threshold θ, if the similarity function $\sigma(s,t) \geq \theta$ in the first k characters we can stop the computation of the similarity of this pair, saving computation and add to our dataset with resulting pairs Dr, also shown in the Algorithm 1.

Algorithm 1 MFKC Similarity Joins

```
1: GoodPairs = {s_1;t_1,...,s_n;t_n}|⟨s,t ∈ ∑*⟩
2: hs = {e_1,e_2,...,e_n}|e_i = ⟨c,f(c,s)⟩
3: ht = {e_1,e_2,...,e_n}|e_i = ⟨c,f(c,t)⟩
4: i, freq ∈ ℕ*
5: procedure MFKC(D_s, D_t, θ, k)
6:     for all s ∈ D_s do (in parallel)
7:         hs = h(s,k)
8:         for all t ∈ D_t do (in parallel)
9:             ht = h(t,k)
10:            if (hs_1(k)+|t|)/(|s|+|t|) < θ then
11:                Next t ∈ D_t
12:            end if
13:            if |hs ∩ ht| = 0 then
14:                Next t ∈ D_t
15:            end if
16:            for all c_freq ∈ hs ∩ ht do
17:                if i ≥ k then
18:                    Next t ∈ D_t
19:                end if
20:                freq = freq + c_freq
21:                σ_i = freq/(|s|+|t|)
22:                if σ_i ≥ θ then
23:                    GoodPairs.add(s,t)
24:                    Next t ∈ D_t
25:                end if
26:                i = i + 1
27:            end for
28:        end for
29:    end for
30:    return GoodPairs
31: end procedure
```

4 Correctness and Completeness

In this section, we prove formally that our MFKC is both correct and complete.

- We say that an approach is correct if the output O it returns is such that $O \subseteq R(D_s, D_t, \sigma, \theta)$.
- Approaches are said to be complete if their output O is a superset of $R(D_s, D_t, \sigma, \theta)$, i.e., $O \supseteq R(D_s, D_t, \sigma, \theta)$.

Our MFKC consists of three nested filters, each of which creates a subset of pairs, i.e. $\mathcal{A} \subseteq \mathcal{L} \subseteq \mathcal{N} \subseteq D_s \times D_t$. For the purpose of clearness, we name each filtering rule:

$$R_1 \triangleq \frac{h_1(s,k)k + |t|}{|s| + |t|} < \theta$$

$$R_2 \triangleq |h(s,k) \cap h(t,k)| \neq 0$$

$$R_3 \triangleq \sigma(s,t) \geq \theta$$

Each subset of our MFKC can be redefined as $\mathcal{N} = \{\langle s,t\rangle \notin D_s \times D_t : R_1$, $\mathcal{L} = \{\langle s,t\rangle \in D_s \times D_t : R_1 \wedge R_2$, and $\mathcal{A} = \{\langle s,t\rangle \in D_s \times D_t : R_1 \wedge R_2 \wedge R_3$. We then introduce $\mathcal{A}^*$ as the set of pairs whose similarity score is more or equal than the threshold θ.

$$\mathcal{A}^* = \{\langle s,t\rangle \in D_s \times D_t : \sigma(s,t) \geq \theta\} = \{\langle s,t\rangle \in D_s \times D_t : R_3\} \tag{17}$$

Theorem 2. *Our MFKC filtering algorithm is correct and complete.*

Proof (Theorem 2). Proving Theorem 2 is equivalent to showing that $\mathcal{A} = \mathcal{A}^*$. Let us consider all the pairs in A. While our MFKC's correctness follows directly from the definition of $\mathcal{A}$, it is complete iff none of pairs discarded by the filters actually belongs to A^*. Assuming that the hashes are sorted in a descending way according the frequencies of the characters, therefore the first element of each hash has the highest frequency. Therefore, once we have

$$\frac{h_1(s,k)k + |t|}{|s| + |t|} < \theta,$$

the pair of strings s and t can be discarded without calculating the entire similarity. When rule R_3 applies, we have $\sigma(s,t) < \theta$, which leads to $R_3 \Rightarrow R_1$. Thus, set $\mathcal{A}$ can be rewritten as:

$$\mathcal{A} = \{\langle s,t\rangle \in D_s \times D_t : R_2 \wedge R_3\} \tag{18}$$

We are given two strings s and t and the respective hashes $h(s,k)$ and $h(t,k)$, the intersection between the characters of these two hashes is a empty set. Therefore, there is no character matching, which implies that s and t cannot be considered to have a similarity score greater than or equal to threshold θ:

$$\mathrm{h}(s,k) \cap \mathrm{h}(t,k) = \emptyset \Rightarrow \sigma(s,t) = 0$$

When the rule R_3 applies, we have $\sigma(s,t) < \theta$, which leads to $R_3 \Rightarrow R_2$. Thus, set $\mathcal{A}$ can be rewritten as:

$$\mathcal{A} = \{\langle s,t\rangle \in D_s \times D_t : R_3\} \tag{19}$$

which is the definition of $\mathcal{A}^*$ in Equation (17). Therefore, $\mathcal{A} = \mathcal{A}^*$.

Time complexity In order to calculate the time complexity of our MFKC, firstly we considered the most frequent K characters from a string. The first step is to sort the string lexically. Then, we can reach a linear complexity after this sort, because the input with highest occurrences can be achieved with a linear time complexity. The first string can be sorted in $O(nlogn)$ and second string in $O(mlogm)$ times, as some classical sorting algorithms such as merge sort [3] and quick sort [4] that work in $O(nlogn)$ complexity. Thus, the total complexity is $O(nlogn) + O(mlogm)$, resulting in $O(nlogn)$ as upper bound in the worst case.

5 Evaluation

The aim of our evaluation is to show that our work outperforms the naive approach and the parallel implementation has a performance gain in large datasets with size greater than 10^5 pairs. A considerable number of pairs reach the threshold θ before reaching the last k most frequent character, that also is a demonstration about how much computation was avoided. Instead of all k's we just need $k - n$ where n is the n^{th} most frequent character necessary to reach the threshold θ. An example to show the efficiency of each filter can be found at Figures 1 and 1(a), where 10,273,950 comparisons from DBpedia+LinkedGeoData were performed and Performance Gain $(PG) = Recall(\mathcal{N}) + Recall(\mathcal{L})$. The recall[2] can be seen in Figure 2(c). This evaluation has the intention to show results of experiments on data from DBpedia[3] and LinkedGeoData[4]. We considered pairs of labels in order to do the evaluation. We have two motivations to chose these datasets: (1) they have been widely used in experiments pertaining to Link Discovery (2) the distributions of string sizes between these datasets are significantly different [1]. All runtime and scalability experiments were performed on a Intel Core i7 machine with 8GB RAM, a video card NVIDIA NVS4200 and running Ms Windows 10.

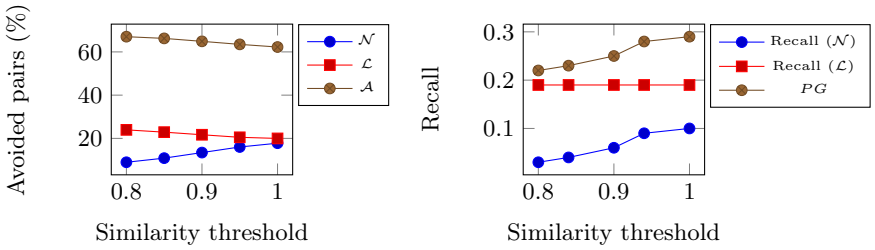
5.1 Parallel implementation

Our algorithm contains parallel code snippets with which we perform a load and balance of the data among CPU/GPU cores when available. To illustrate this

[2] Depicting DBPedia-Yago results. The YAGO was added to bring a reinforcement to our evaluations, due to the fact of this dataset have been widely used in experiments pertaining to Link Discovery. We also considered evaluations on recall, precision, and f-score.

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

[4] `http://linkedgeodata.org/`



(a) Avoided pairs(10,273,950). (b) Recall.

Fig. 1. Avoided pairs and recall.

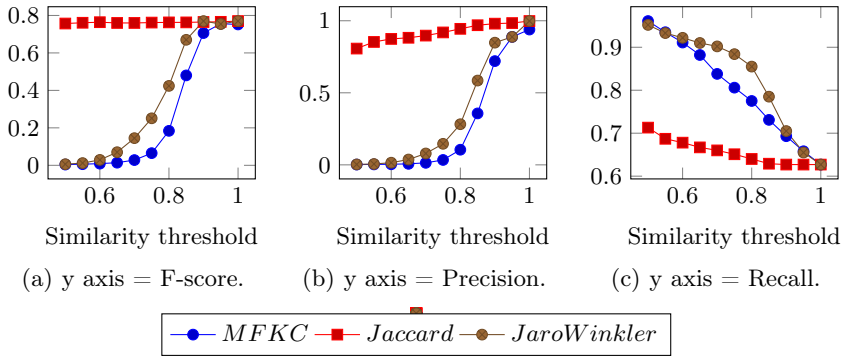
part of our idea, we can state: Given a two datasets S, T, that contains all the strings to be compared. Thus, make a Cartesian product of the strings $S \times T$, where each pair is the processed separately in threads that are spread among CPU/GPU cores. Thus, we process the each comparison in parallel. The parallel implementation works better in large datasets with size more than 10^5, that was more than one time faster than the approach without parallelism and two times faster than the naive approach as shown in Figures 3(a) to 3(c).

5.2 Runtime Evaluation

The evaluation in Figures 3(a) and 3(b) shows that all filter setup outperform the naive approach, and the parallel approach does not suffer significant changes related to the runtime according to the size of the dataset, as show Figure 3(c). The experiments related to the variance of k, also were considered, as show in Figure 3(d), the runtime varies according the size of k, indicating the influence of k with values from 1 to 120 with $1,001,642$ comparisons. The performance (run-time) was improved as shown in Figures 3(a) and 3(b) and according the recall with a performance gain of 26.07% as shown in Figure 1(a). The time complexity is based on two sort process O(n log n) + O(m log m) resulting in O(n log n) as a upper bound in the worst case.

5.3 Scalability Evaluation

In the experiments (see Figures 3(c), 3(e) and 3(f)), we looked at the growth of the runtime of our approach on datasets of growing sizes. The results show that the combination of filters ($\mathcal{N}$+$\mathcal{L}$+$\mathcal{A}$) is the best option for datasets of large sizes. This result holds on both DBpedia and LinkedGeoData, so our approach can be used on large datasets and achieves acceptable run-times. We also can realize the quantity of avoided pairs in each combination of filters in Figure 1, that consequently brings a performance gain. We looked at experiments with runtime behavior on a large dataset with more than 10^6 labels as shown in Figure 3(b). The results suggest that the runtime decreases according to the threshold θ



(a) y axis = F-score. (b) y axis = Precision. (c) y axis = Recall.

Fig. 2. Precision, Recall and F-Measure.

increment. Thus, one more point showing that our approach is useful on large datasets, where can be used with high threshold values for link discovery area. About our parallel implementation, Figure 3(c) shows that our GPU parallel implementation works better on large datasets with size greater than 10^5.

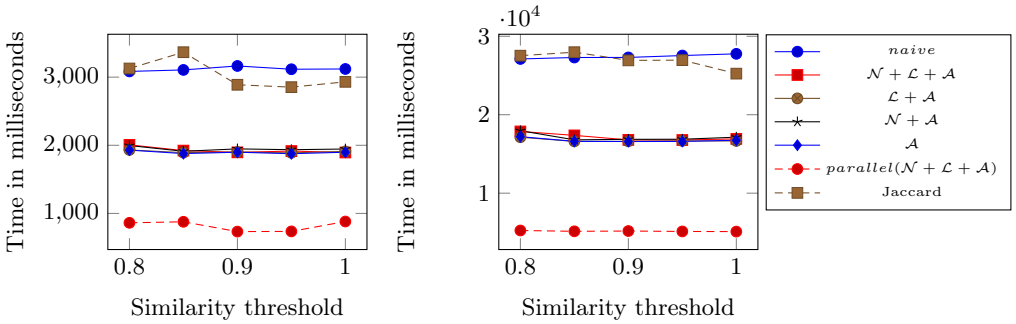
5.4 Comparison with existing approaches

Our work overcomes the naive approach [8], thus, in order to show some important points we compare our work not only with the state of the art, but with popular algorithms such as Jaccard Index [5]. As shown in Figures 3(c), 3(e) and 3(f), our approach outperforms not only the naive approach, but also Jaccard Index. We show that the threshold θ and k have a significant influence related to the runtime. The naive approach present some points to consider, among them, even if the naive approach states that they did experiments with $k = 7$, the naive algorithm was designed for only $k = 2$, there are some cases where $k = 2$ is not enough to get the similarity level expected, i.e. $s = mystring1$ and $t = mystring2$ limiting $k = 2$, we will have $\sigma_2(s, t) = 0.2$, showing that the similarity is very low, but when $k = 8$, the similarity is $\sigma_8(s, t) = 0.8$ showing that sometimes we can lose a good similarity case limiting $k = 2$. Our work fix all these problems and also has a better runtime, as show Figure 3(a), Figure 3(b) and Figure 3(c).

An experiment with labels from DBpedia and Yago[5] shows that the f-score indicates a significant potential to be used with success as a string similarity comparing with Jaccard Index and Jaro Winkler, as Figures 2(a) to 2(c) shows.

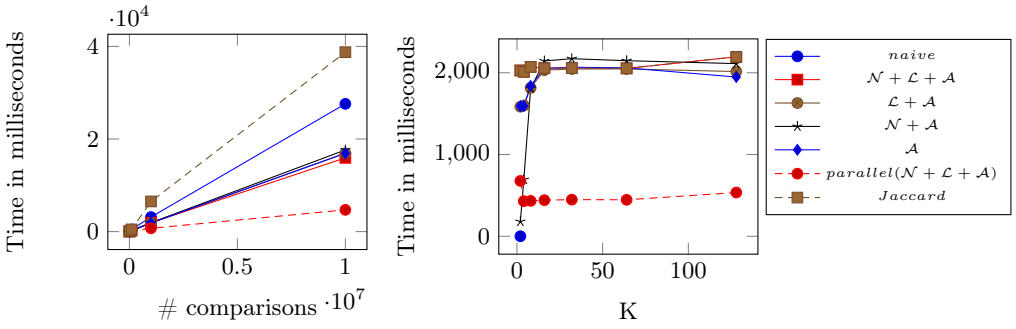
To summarize the key features that makes our approach outperform the naive approach are the following: We use more than two K most frequent characters in our evaluation, our run-time for more than 10^7 comparisons is shorter (27,594 against 16,916) milliseconds, we do have a similarity threshold, allowing us to

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



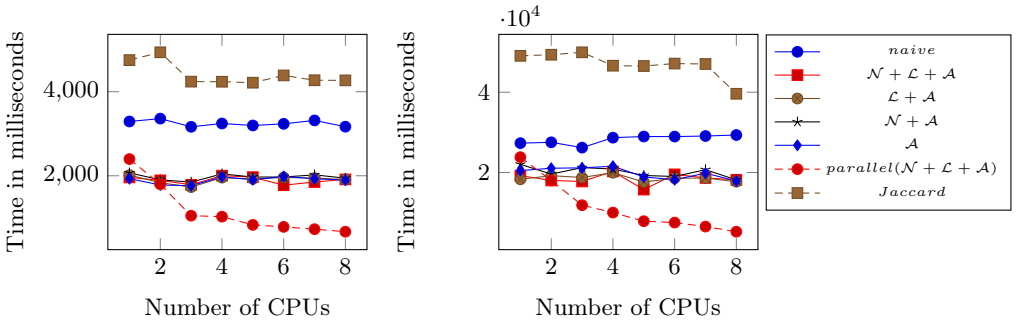
(a) Runtime 1,001,642 comparisons.

(b) Runtime 10,273,950 comparisons.



(c) The parallel approach improves the performance for more than 10^5*comparisons*.

(d) Runtime k most frequent characters, with values of k from 1 to 120, over $1,001,642$ comparisons, $\theta = 0.95$.



(e) CPU Speedup of algorithm $(1,001,642$ comparisons).

(f) CPU Speedup of algorithm$(10,273,950$ comparisons).

Fig. 3. Run-time experiments results.

discard comparisons avoiding extra processing, and a parallel implementation, making our approach scalable. Jaccard does not show significant changes varying the threshold, MFKC and Jaro Winkler present a very similar increase of the f-score varying the threshold.

6 Conclusion and Future work

We presented an approach to reduce the computation runtime of similarity joins using the Most Frequent k Characters algorithm with a sequence of filters that allow discarding pairs before computing their actual similarity, thus reducing the runtime of computation. We proved that our approach is both correct and complete. The evaluation shows that all filter setup outperform the naive approach. Our parallel implementation works better in larger datasets with size greater than 10^5 pairs. It is also the key to developing systems for Record Linkage and Link Discovery in knowledge bases. As future work, we plan to integrate it in link discovery applications for the validation of equivalence links. The source code is free and available online[6]. Acknowledgments available on footnotes[7].

References

1. K. Dreßler and A.-C. N. Ngomo. On the efficient execution of bounded jaro-winkler distances. 2014.
2. J. Euzenat, P. Shvaiko, et al. *Ontology matching*, volume 333. Springer, 2007.
3. H. H. Goldstine and J. von Neumann. *Planning and coding of problems for an electronic computing instrument*. Institute for Advanced Study, 1948.
4. C. A. Hoare. Quicksort. *The Computer Journal*, 5(1):10–16, 1962.
5. P. Jaccard. The distribution of the flora in the alpine zone. *New phytologist*, 11(2):37–50, 1912.
6. V. I. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710, 1966.
7. R. Rivest. The MD5 message-digest algorithm. 1992.
8. S. E. Seker, O. Altun, U. Ayan, and C. Mert. A novel string distance function based on most frequent k characters. *IJMLC*, 4(2):177–182, 2014.
9. S. E. Seker and C. Mert. A novel feature hashing for text mining. *Journal of Technical Science And Technologies*, 2(1):37–40, 2013.
10. P. Shvaiko and J. Euzenat. Ontology matching: State of the art and future challenges. *TKDE*, 25(1):158–176, 2013.
11. T. Soru and A.-C. Ngonga Ngomo. Rapid execution of weighted edit distances. In *Proceedings of the Ontology Matching Workshop*, 2013.
12. A. Valdestilhas, N. Arndt, and D. Kontokostas. DBpediaSameAs: An approach to tackle heterogeneity in dbpedia identifiers.
13. C. Yan, X. Zhao, Q. Zhang, and Y. Huang. Efficient string similarity join in multi-core and distributed systems. *PLOS ONE*, 12(3):1–16, 03 2017.

[6] `https://github.com/firmao/StringSimilarity/`

[7] Acknowledgments to *CNPq* Brazil under grants No. 201536/2014-5 and H2020 projects SLIPO (GA no. 731581) and HOBBIT (GA no. 688227) as well as the DFG project LinkingLOD (project no. NG 105/3-2), the BMWI Projects SAKE (project no. 01MD15006E) and GEISER (project no. 01MD16014).

Semantic Interactive Ontology Matching: Synergistic Combination of Techniques to Improve the Set of Candidate Correspondences

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

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

Abstract. Ontology Matching is the task of finding a set of entity correspondences between a pair of ontologies, i.e. an alignment. It has been receiving a lot of attention due to its broad applications. Many techniques have been proposed, among which the ones applying interactive strategies. An interactive ontology matching strategy uses expert knowledge towards improving the quality of the final alignment. When these strategies are based on the expert feedback to validate correspondences, it is important to establish criteria for selecting the set of correspondences to be shown to the expert. A bad definition of this set can prevent the algorithm from finding the right alignment or it can delay convergence. In this work we present techniques which, when used simultaneously, improve the set of candidate correspondences. These techniques are incorporated in an interactive ontology matching approach, called ALINSyn. Experiments successfully show the potential of our proposal.

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

1 Introduction

Ontology matching seeks to discover correspondences between entities of different ontologies [1]. Ontology matching can be processed manually, semi-automatically or automatically [1]. Among the semi-automatic approaches, the ones that follow an interactive strategy stand out, considering the knowledge of domain experts through their participation [2]. The involvement of a domain expert is not always possible, as it is an expensive, scarce and time-consuming resource. However, when possible, better results have been achieved compared with automatic approaches.

An expert can be involved by giving his feedback to a correspondence, indicating whether or not it belongs to the alignment. Therefore, defining the set

of correspondences to show to the expert is one of the problems of these interactive techniques. If this set is not well defined, the final alignment may be imprecise or incomplete, or convergence to a good alignment can be delayed. Therefore, the scientific problem addressed in this paper is how to improve the set of correspondences to receive expert feedback.

This paper proposes ALINSyn, an approach that uses two techniques – a semantic and a structural – for the improvement of a given set of candidate correspondences. The semantic technique works by temporarily removing correspondences from the set of candidate correspondences. The structural technique interactively places part of the correspondences taken by the semantic technique back in the set of candidate correspondences. ALINSyn uses techniques used in the ALIN [13] system, that participated in OAEI 2016.

To evaluate ALINSyn, we defined ALINBasic, a basic ontology matching algorithm that generates and use a set of candidate correspondences to do the matching. Each of the two ALINSyn techniques was added to ALINBasic in order to modify the set of candidate correspondences generated by it, and the obtained alignments were compared. ALINSyn was also compared to state-of-the-art interactive ontology matching systems, showing the potential of our proposal.

This paper is structured as follows: Section 2 describes interactive ontology matching, Section 3 describes the ALINBasic algorithm, section 4 describes ALINSyn approach, by explaining its two steps, in section 5 the evaluation of the approach is made and the section 6 is the conclusion.

2 Interactive Ontology Matching

An ontology O is represented as a labeled graph G = (V, E, vlabel, elabel). The set of vertices V contains ontology entities such as concepts and properties. Edges in E ($E \subseteq V \times V$) represent structural relationships between entities. The edge labeling function elabel, which maps an edge (v, v) $\in$ E to a subset of the set SL of structural labels, which in turn specify the nature of the structural relationships between entities (e.g., subclassOf). Let LL denote the set of lexical labels associated with entities (e.g., name, documentation). Finally, the vertex labeling function, vlabel : $V \times LL \rightarrow$ String, maps a pair (e, l) $\in V \times LL$ to a string corresponding to the value of the lexical label l (e.g., name) associated with the entity e [3].

Given two ontologies O and O', an ontology matching is the process that aims to finding a set of correspondences (e, e'), where e and e' are entities in O and O', respectively. Interactive ontology matching takes advantage of user feedback to perform ontology matching.

Within the set of all possible correspondences between the entities of two ontologies, in the context of the interactive ontology matching, we distinguish two types of correspondences:

– Candidate correspondences are those possible correspondences that have been selected to be presented to the expert but have not yet received decision,

– Classified correspondences are those possible correspondences that have been selected to be presented to the expert and have received decision.

There are similarity measures, denoted sim, which map the possible correspondence (e, e') $\in O \times O'$ to a real number in [0, 1].

According to Meilicke and Stuckenschmidt [4], ontology matching algorithms that are based on the analysis of entity names usually have two phases:

– In the first phase, there is the creation of a set of candidate correspondences. To reduce the need to classify all possible correspondences (all pairs of entities) between two ontologies as belonging or not to alignment, the algorithm selects a subset called set of candidate correspondences;
– In the second phase, each correspondence in the set of candidate correspondences is classified by the ontology matching algorithm. In an interactive strategy, at least part of these correspondences is classified by the expert, and the other part can be classified by some automatic technique.

3 ALINBasic Algorithm

When the ontology matching is done interactively, we have two quality measures that are conflicting: the number of interactions with the expert and the quality of the generated alignment. It is interesting that a technique to be used in an algorithm of ontology matching can improve one of these qualities without worsening the other in an accentuated way. That is, to decrease the number of interactions without decreasing proportionally the quality of the generated alignment, or to increase the quality of the generated alignment without increasing proportionally the number of interactions with the expert.

In this paper two techniques will be presented, which used alone, cannot increase one of the qualities without considerably worsening the other. The first one, the semantic technique, decreases the number of interactions with the expert, but greatly decrease the quality of the generated alignment. The other, structural technique, enhances the quality of the generated alignment, but increasing a lot the number of interactions with the expert. But when used together, they can mitigate the disadvantages of each other, reducing the number of interactions without dramatically decreasing the quality of the generated alignment.

To evaluate the results of the two proposed techniques, three algorithms will be compared. An algorithm without the inclusion of any of the two techniques, called ALINBasic, a second algorithm, with the inclusion of the semantic technique, called ALINSem, and a third one with the inclusion of both the semantic and structural techniques, called ALINSyn. The two techniques are included in the algorithms as steps of these algorithms, so ALINSem is equivalent to the ALINBasic algorithm plus a semantic step that implements the semantic technique, and the ALINSyn algorithm is equivalent to the ALINSem algorithm plus a structural step that implements the structural technique.

The ALINBasic algorithm has two phases, as described by Meilicke and Stuckenschmidt [4]. The first phase selects candidate correspondences to be presented to the user. The second phase presents the selected candidate correspondence to the user and assigns them to the classified correspondences. Hence, in the end there are no candidate correspondences left.

In the phase of generating the candidate correspondences, only class correspondences, not property correspondences, are chosen, therefore, the ALINBasic algorithm finds only class correspondences.

The first phase of ALINBasic (Algorithm 1) will use the stable marriage algorithm with size list limited to 1 [5][6], where the pair will be formed by classes of the two ontologies to be aligned. Correspondences will be ordered by decreased similarity.

The stable marriage algorithm will be executed six times, each time with a different similarity metric (Jaccard, Jaro-Winkler, n-Gram, Wu-Palmer, Jiang-Conrath and Lin) and the result of the six executions will form a set of correspondences by the union of the six formed sets (Steps 1 to 4 of Algorithm 1). The process of selecting the similarity metrics was based on two criteria: available implementations and the result of these metrics in assessments, such as those carried out in [7] and [8]. Wu-Palmer, Jiang-Conrath and Lin are metrics that require a taxonomy to be computed [7], this taxonomy being provided, in this algorithm, by Wordnet.

From the set of correspondences formed by the union of the six sets all correspondence whose classes have exactly the same name will be classified as true (Step 5 of Algorithm 1). The correspondences selected by the running of stable marriage algorithm and not automatically classified will be the candidate correspondences (Step 6 of Algorithm 1).

Algorithm 1 Candidate correspondence generation

Input: Two ontologies to be aligned
Output: Candidate correspondences

1: **for** Each one of the similarity metrics: Jaccard, Jaro-Winkler, n-Gram, Wu-Palmer, Jiang-Conrath and Lin **do**
2: Run stable Marriage Algorithm forming the set A_{sim} (being sim the corresponding similarity metric)
3: **end for**
4: Let A = $A_{Jaccard} \cup A_{Jaro\text{-}Winkler} \cup A_{n\text{-}Gram} \cup A_{Wu\text{-}Palmer} \cup A_{Jiang\text{-}Conrath} \cup A_{Lin}$
5: Let B = Correspondences, from A, automatically classified as true by the fact that their entities have the same name
6: Set of candidate correspondences = A - B

Then begins the classification phase of the candidate correspondences of the ALINBasic. At this phase all the candidate correspondences will be presented to the expert to receive his feedback.

For this, the concept of interaction with the expert will be used. An interaction with the expert corresponds to a question asked about at most three correspondences, as long as they pair-wisely have at least one of the entities in common. This is compliant with the OAEI definition [10]. For example, if the following correspondences are shown to the expert at the same time (ConferenceChair,Chair), (Chairman,Chair) and (Chairman,AssociatedChair), they will be counted as only one interaction since each correspondence has at least one entity of another correspondence. The number of interactions will be used as a comparison criterion between the various executions shown in this paper.

The ALINBasic algorithm can be seen in Algorithm 2.

Algorithm 2 ALINBasic

Input: Two ontologies to be aligned
Output: Alignment between the two ontologies
1: Run candidate correspondence generation (Algorithm 1)
2: **for** Each candidate correspondence **do**
3: Receive feedback (the candidate correspondence is transformed to classified correspondence)
4: **end for**

4 ALINSyn Algorithm

4.1 Improving the Set of Candidate Correspondences

The objective of the ALINSyn algorithm is to decrease the number of interactions with the expert without decreasing in the same proportion the quality of the generated alignment. To achieve this objective, two steps, one semantic step and one structural step, are added to the ALINBasic algorithm to improve the set of candidate correspondences.

We first introduce another type of correspondence:

- Temporarily suspended correspondences are correspondences that are no longer candidate correspondences because of the semantic step. These correspondences can once again be candidate correspondences after the structural step.

The semantic step transforms some candidate correspondences to temporarily suspended correspondences. The structural step can transform some temporarily suspended correspondences to candidate correspondences again.

At the end of the non-interactive phase, by the use of the semantic step, all candidate correspondences that are not semantically equivalent will be transformed to temporarily suspended correspondences. In the interactive phase, by the use of the structural step, after each interaction with the expert, the expert's feedback can transform temporarily suspended correspondences in candidate correspondences if they have a particular structural relationship with a candidate correspondence that received positive feedback.

4.2 Semantic Step

The action of this step is to transform all candidate correspondences with semantically different entity names to temporarily suspended correspondences. The step will be added to the ALINBasic algorithm at the end of the generation phase.

The semantic step uses Wordnet. Wordnet consists of synonym sets called synsets [9]. A synset denotes a group of terms with the same meaning. The same term may appear in various synsets, as long as it has several meanings.

Comparison of entity names A head noun of a phrase is a noun to which all other terms are dependent [11]. Only correspondences relating entities whose name head nouns are in the same Wordnet synset will remain in the set of candidate correspondences after the semantic step. Before comparing the two entity names, a pre-processing step is necessary in order to extract the correct terms to be compared. An entity name can be atomic or composed. In the latter case, our approach searches for the head noun, and only this head noun will be used to compare the two entities. The rule we used for detection of head noun can be summarized as follows:

1. If the name contains a preposition (e.g. HeadOfDepartment) then the head noun is the token before the preposition.

2. Otherwise the head noun is the last token in the name.

Algorithm 3 Semantic step

Input: Candidate correspondences

Output: Temporarily suspended correspondences (ex-candidate correspondences)

1: **for** Each candidate correspondence **do**
2: Choose the head noun of each entity of the name of the correspondence
3: Put the head noun of each name in the canonical form
4: **if** The two head nouns are not in the same wordnet synset **then**
5: Transform the candidate correspondence to temporarily suspended correspondence
6: **end if**
7: **end for**

Example of the semantic step The semantic step can be seen in the Algorithm 3. To illustrate the semantic step we assume that we have the candidate correspondences selected by Algorithm 1 shown in Table 1. The first correspondence to be analyzed will be (Author, Regular_Author) (step 1 of Algorithm 3). The head noun of Author is Author, since it has only one word. The head noun for Regular_Author is Author, because it does not have a preposition and the last word is Author (step 2). The two head nouns are already in canonical form (step 3) and as they are the same word they are in the same synset, so they are not transformed to temporarily suspended correspondences.

Table 1. Correspondences before and after the semantic step, and after the first run of structural step

e	e'	before semantic step	after semantic step	after the first run of structural step
Author	Regular_author	Candidate correspondence	Candidate correspondence	Candidate correspondence
Chairman	Chair	Candidate correspondence	Temporarily suspended correspondence	Temporarily suspended correspondence
Co-author	Regular_author	Candidate correspondence	Temporarily suspended correspondence	Candidate correspondence
Paper	Paper	Candidate correspondence	Candidate correspondence	Candidate correspondence
Paper_Abstract	Abstract	Candidate correspondence	Candidate correspondence	Candidate correspondence
Person	Person	Candidate correspondence	Candidate correspondence	Classified as true
Subject_Area	Abstract	Candidate correspondence	Temporarily suspended correspondence	Temporarily suspended correspondence
Subject_Area	Program_Committee	Candidate correspondence	Temporarily suspended correspondence	Temporarily suspended correspondence

The second correspondence in the table is the correspondence (Chairman,Chair) (step 1). Chairman is considered a word because a term is only divided into words if it has hyphen, white space or is in camelcase (step 2). Since the two are in the canonical form (step 3) of the word their synsets are compared in Wordnet, and they are different. It is important to note that the most common meanings of words are searched for in wordnet, so Chair is the object of sitting and not Boss. Therefore this correspondence will be transformed to temporarily suspended correspondences (step 5).

The result after following these steps for all correspondences is shown in Table 1, in the column 'after the semantic step'.

Algorithm 4 ALINSyn

Input: Two ontologies to be aligned
Output: Alignment between the two ontologies
1: Run candidate correspondence generation (Algorithm 1)
2: Run semantic step (Algorithm 3)
3: **for** Each candidate correspondence **do**
4: Receive feedback (the candidate correspondence is transformed to classified correspondence)
5: Run structural Step (Algorithm 5)
6: **end for**

With the inclusion of the semantic step, the algorithm will be called ALINSem. As an illustration, this algorithm is the same as the algorithm ALINSyn (Algorithm 4) without the inclusion of step 5 (Run structural step). The results of ALINSem will be compared to the results of ALINSyn with the objective of verifying if the combined use of the semantic step and the structural step improves the result achieved by the use of the semantic step alone.

Algorithm 5 Structural Step

Input: Temporarily suspended Correspondences, Classified correspondences
Output: Candidate Correspondences (ex-temporarily suspended correspondences)

```
for Each temporarily suspended correspondence do
    if The two classes of the temporarily suspended correspondence are subclasses of classes of a correspondence classified as true then
        Transform the temporarily suspended correspondence to candidate correspondence
    end if
end for
```

4.3 Structural Step

When only the semantic step is applied, experiments showed that the number of interactions with the expert were reduced, i.e. convergence was reached faster, however the final alignment lost in quality. This is because some true correspondences have been taken from the set of candidate correspondences because of semantic step. The main goal of the structural step is to recover part of the quality lost through the use of the semantic step by transforming some temporarily suspended correspondences again to candidate correspondences.

At each iteration, all temporarily suspended correspondences that are formed by subclasses of the classes of the correspondences that received positive feedback from the expert are transformed again to candidate correspondences. Tests were performed again using the two techniques, which showed that the use of both techniques makes the number of interactions decrease considerably, but with a much lower quality loss, in relation to the results obtained with the ALINBasic algorithm. The structural step can be seen in Algorithm 5.

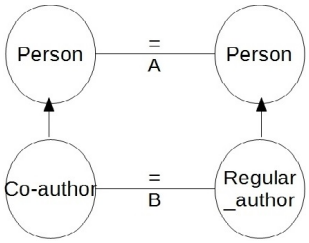


Fig. 1. Correspondences with classes that are subclasses of other correspondence classes

To illustrate the technique let us assume the situation described in Figure 1, where Co_author is a subclass of Person in the cmt ontology and Regular_author is a subclass of Person in the Conference ontology. Let us assume that the correspondence A (Person, Person) is a candidate correspondence and correspondence B (Co_author, Regular_author) is a temporarily suspended correspondence. If the

correspondence A receive positive feedback, the correspondence B by having its classes that are subclasses of the classes of A is transformed to candidate correspondence. The result of the structural step can be seen in Table 1 in the column 'after the first run of the structural step'. With the inclusion of the structural step in the interactive phase, the algorithm is called ALINSyn and can be seen in the Algorithm 4.

5 Evaluation Overview and Designed Analysis

The goal of the ALINSyn approach is to reduce the number of interactions with the expert without greatly diminishing the quality of the generated alignment. Thus a first research question is:

RQ1: Does the semantic step allow the ontology matching strategy to decrease the number of interactions with the expert? This question is answered with the use of the semantic step in the ALINBasic algorithm, as we see in the section "Analysis of the Results", which shows that the number of interactions with the expert has been reduced, but with a great drop in quality. That is why it is important to address other research questions.

RQ2: Can the expert feedback reduce the quality loss by the use of the semantic step?

RQ3: Does the use of both, semantic step and structural step together, generate an alignment with quality and number of interactions compatible with the state of the art proposals?

5.1 Conference dataset

Results obtained in the interactive matching of OAEI 2016 using the conference dataset were used to compare with the state of the art.

The OAEI interactive track is performed with percentages of expert correctness, from 70% to 100%. This paper has taken into consideration, for the evaluation of the execution of the ALINSyn and of other tools, 100% of correctness by the expert.

5.2 Analysis of the Results

After using the semantic step the results presented in Table 2 (ALINSem row) were reached, which shows that the use of the semantic step decreases the number of expert interactions, which responds to 'RQ1: Does the semantic step allow the ontology matching strategy to decrease the number of iterations with the expert ?', but there has been a sharp drop in quality, which shows the need to answer the question 'RQ2: Can the expert feedback reduce the quality loss by using the semantic step?'.

The recovery in the quality of the generated alignment was attempted by the use of structural step. After the inclusion of this new step the results shown in

Table 2 (ALINSyn row) were reached. That shows that the goal of the ALINSyn was achieved using the two techniques. The number of interactions with the expert decreased greatly, from 619 to 219, with the quality decreasing proportionally much less, the f-measure was from 0.79 to 0.75, what responds to RQ2: Can the expert feedback reduce the quality loss by the use of the semantic step ?. The result achieved is due to the combined effect of the joint use of the two techniques.

If we use only the semantic step we have a good decrease in the number of interactions with the expert, but with a sharp drop in quality. The subsequent use of the structural step, interactively, causes some of the lost quality to be recovered.

If we use only the structural step, without using the semantic step before, with all possible correspondences, not only the temporarily suspended correspondences, we would have an increase in quality, but a large number of correspondences would be added to the set of candidate correspondences, which would make the number of interactions with the expert too large (Table 2, ALINStr row). The transformation of candidate correspondences into temporarily suspended correspondences, through the semantic step, and the search, by the structural step, only among the temporarily suspended correspondence reduces the search space, which means that the number of interactions with the expert do not go up explosively.

The combined use of the two techniques results in a more balanced result, with a reduction in the number of interactions without a big loss of quality (Table 2, ALINSyn row).

Table 2. Comparison between different matching executions

	NI	Precision	F-measure	Recall
ALINBasic	619	0.92	0.79	0.70
ALINSem	152	0.90	0.69	0.57
ALINStr	3539	0.93	0.84	0.78
ALINSyn	219	0.91	0.75	0.65

5.3 Comparison among Tools that Participated in the OAEI Interactive Conference Track

OAEI provides a comparison among tool performance in the ontology matching process each year, and one of the ontology groups used is the conference dataset used in this paper [12].

Table 3 shows a comparison of some tools that participated in the OAEI 2016 interactive conference track. NI means number of interactions. In each interaction there can be up to three questions. "%" is the ratio of the number of interactions to the number of possible correspondences among all the alignments of the conference dataset.

Table 3 compares the performance of ALINSyn with some interactive tools that participated in OAEI 2016, with the expert hitting 100% of the answers in relation to the conference dataset. The use of the techniques shown in this work generates a high quality alignment, in cases where the expert does not make errors, what responds to 'RQ3: Does the use of the two techniques, semantic step and structural step together, generate an alignment with quality and number of interactions compatible with the state of the art ?'. The use of the two techniques combined puts ALINSyn among the best tools in the evaluation of OAEI 2016, when the expert hits 100% of the interactions.

Table 3. Comparison between some OAEI 2016 conference dataset interactive tracking tools and ALINSyn

	Number of questions	NI	%	Precision	F-measure	Recall
AML	270	271	0.215	0.912	0.799	0.711
ALINSyn	483	219	0.174	0.915	0.754	0.652
LogMap	142	142	0.113	0.886	0.723	0.610
XMap	4	4	0.003	0.837	0.681	0.574

6 Conclusion

Progress in information and communication technologies has made a large number of data repositories available, but with a great deal of semantic heterogeneity, which makes it difficult to integrate. A process that has been used to solve this problem is the ontology matching, which tries to discover the existing correspondences between the entities of two distinct ontologies, which in turn structures the concepts that define the data stored in each repository.

This work presented an interactive approach for ontology matching, based on manipulation of the set of candidate correspondences with techniques to decrease the number of interactions with the expert, without greatly reducing the quality of the alignment.

Two techniques were combined, one semantic and the other structural. The goal of the semantic technique was to decrease the number of interactions with the expert. The structural technique came in support of the semantic technique, and its objective was to decrease the quality loss resulting from the decrease in the number of interactions with the expert.

In order to evaluate if the techniques generated a decrease in the number of interactions without significantly lowering the quality, the executions of a basic algorithm with and without the techniques were compared, which showed that the techniques, when combined, reach their goal.

In addition, the quality of the alignment provided by the ALINSyn approach was compared to state of the art tools that have participated in the track of interactive ontology matching in OAEI 2016. The results obtained show that ALINSyn 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, keeping the number of interactions within the range achieved by the other tools.

The third author was partially funding by project PQ-UNIRIO N01/2017 (" 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. J. Euzenat and P. Shvaiko, Ontology Matching - Second Edition, 2. Springer-Verlag, 2013.
2. H. Paulheim, S. Hertling, and D. Ritze, Towards Evaluating Interactive Ontology Matching Tools, Lect. Notes Comput. Sci., vol. 7882, pp. 31-45, 2013.
3. S. Duan, A. Fokoue, and K. Srinivas, One Size Does Not Fit All: Customizing Ontology Alignment Using User Feedback, in Lecture Notes in Computer Science (LNCS), 2010, pp. 177-192.
4. C. Meilicke and H. Stuckenschmidt, A New Paradigm for Alignment Extraction, CEUR Workshop Proc., vol. 1545, pp. 1-12, 2015.
5. D. Gale and L. S. Shapley, College Admissions and the Stability of Marriage, Am. Math. Mon., vol. 69, no. 1, pp. 9-15, 2014.
6. R. W. Irving, D. F. Manlove, and G. OMalley, Stable marriage with ties and bounded length preference lists J. Discret. Algorithms, vol. 7, no. 2, pp. 213-219, 2009.
7. E. G. M. Petrakis, G. Varelas, A. Hliaoutakis, and P. Raftopoulou, Design and Evaluation of Semantic Similarity Measures for Concepts Stemming from the Same or Different Ontologies object instrumentality, Proc. 4th Work. Multimed. Semant., vol. 4, pp. 233-237, 2006.
8. M. Cheatham and P. Hitzler, String similarity metrics for ontology alignment, Lect. Notes Comput. Sci., vol. 8219 LNCS, no. PART 2, pp. 294-309, 2013.
9. F. Lin and K. Sandkuhl, A survey of exploiting WordNet in ontology matching, IFIP Int. Fed. Inf. Process., vol. 276, pp. 341-350, 2008.
10. D. Faria, Using the SEALS Client s Oracle in Interactive Matching, 2016. [Online]. Available: https://github.com/DanFaria/OAEI_SealsClient/blob/master/OracleTutorial.pdf
11. O. Svab-Zamazal and V. Svatek, Analysing ontological structures through name pattern tracking, Lect. Notes Comput. Sci., vol. 5268 LNAI, pp. 213-228, 2008.
12. M. Achichi, M. Cheatham, Z. Dragisic, J. Euzenat, D. Faria, A. Ferrara, G. Flouris, I. Fundulaki, I. Harrow, V. Ivanova, E. Jimenez-Ruiz, E. Kuss, P. Lambrix, H. Leopold, H. Li, C. Meilicke, S. Montanelli, C. Pesquita, T. Saveta, P. Shvaiko, A. Splendiani, H. Stuckenschmidt, K. Todorov, C. Trojahn, and O. Zamazal, Results of the Ontology Alignment Evaluation Initiative 2016, Proc. 11th Int. Work. Ontol. Matching co-located with 15th Int. Semant. Web Conf. (ISWC 2016) Kobe, Japan, Oct. 18, 2016., 2016.
13. J. Silva, F. A. Baião, and K. Revoredo, ALIN Results for OAEI 2016, CEUR Workshop Proc., vol. 1766, 2016.

Exploring the Synergies between Biocuration and Ontology Alignment Automation

David Dearing and Terrance Goan

Stottler Henke Associates, Inc.
1107 NE 45th St, Suite 310
Seattle, WA 98105, USA
{ddearing, goan}@stottlerhenke.com

Abstract. Researchers have long recognized the value trapped in natural language publications and have continued to advance the development of ontologies that can help unleash this value. Among these advances are efforts to apply NLP techniques to streamline the labor-intensive process of scientific literature curation, which encodes relevant information in a form that is accessible to both humans and computers. In this paper, we report on our initial efforts to improve ontology alignment within the context of scientific literature curation by exploiting value within a large corpus of annotated PubMed abstracts. We employ an ensemble learning approach to augment a collection of publicly available ontology matching systems with a matching technique that leverages the word embeddings learned from this corpus in order to more successfully match the concepts of two disease ontologies (MeSH and OMIM). Our experiments show that word embedding-based similarity scores do contribute value beyond traditional matching systems. Our results show that the performance of an ensemble trained on a small number of manually reviewed mappings is improved by their inclusion.

Keywords: Ontology Matching Ensembles, Word Embeddings, Biocuration.

1 Introduction

Technological advancements have given rise to an explosion in the rate that biomedical data is generated. The incredible volume of data now far exceeds the ability of researchers to capitalize on it. This is due, in large part, to the vagaries of the natural languages in which that data is published for consumption by human readers. The wide variety of lexical forms employed in the research literature present persistent challenges for both humans and computers in finding, assessing, and assimilating relevant data.

The research community has long recognized the value trapped in natural language publications and has continued to advance the development of ontologies that can mitigate the challenges posed by natural language. Today, ontologies are a critical foundation for emerging technologies that seek to better inform and accelerate biomedical research. Notable among recent advances are efforts to apply Natural Language Processing (NLP) techniques to streamlining the labor-intensive processes of biocuration and systematic scientific reviews.

Biocuration involves the interpretation, representation, and integration of information relevant to biology into a form that is accessible to both humans and computers. This process results in databases or knowledgebases (e.g., UniProt [1], NCBI Database Resources [2], and the Rat Genome Database (RGD) [3]) that assimilate the scientific literature as well as large data sets. Biocuration efforts range in both approach and scope, but they are increasingly supported by automated tools that facilitate information triage and tagging [4, 5].

Similar to biocuration is the systematic review: a literature review that gathers and analyzes research literature according to a structured methodology and guided by one or more specific research questions. The aim of systematic review is to produce an exhaustive summary of current literature relevant to those research questions. Sometimes a systematic review is simply an instance of a biocuration effort without sufficient resources to codify the collected knowledge [6]. As with biocuration, there are increasing efforts to employ natural language processing and other artificial intelligence methods to streamline an expert-driven process that is otherwise very labor intensive [7-10].

Biocuration and systematic review processes (whether manual or automated) are complicated by the applicability of overlapping ontologies that cover a breadth of multispecies knowledge that ranges across biological scales from molecules to populations. Ultimately, the exploitation of numerous (but well-aligned) ontologies will provide a comprehensive landscape of biomedical knowledge that will speed the identification of new hypotheses and avenues of investigation.

In this paper, we report on our initial efforts to improve ontology alignment within the context of scientific literature curation. More specifically, we describe an ensemble learning approach that augments a collection of ontology matching systems with word embeddings generated from an annotated corpus of relevant scientific literature.

The rest of this paper is organized as follows: In the next section, we provide background and discuss related work. In Sections 3 and 4 we describe our experiments, research hypothesis, and results. Finally, in Section 5, we summarize our conclusions and plans for future work, including extensions that support learning from work-centered user interactions.

2 Background and Related Work

The best-performing ontology matching tools all rely on collections of complementary matchers in order to compensate for context-specific weaknesses of each contributing/competing heuristic. The challenge of matcher selection and evidence combination has been addressed in a variety of ways ranging from ad hoc rules and manual settings [11] to ensemble learning methods [12, 13] that utilize machine learning to select and weight contributing matchers. Methods, such as “mapping gain” measurement, are applicable to the related challenge of selecting appropriate background knowledge sources [14].

Background knowledge sources play an important role in the performance of ontology matching tools. While string distance measures and taxonomic structure comparison form the backbone of most tools for ontology matching, it is also widely recognized

that ontologies constructed by independent experts can differ significantly in both organization and lexical features. In these situations, researchers commonly seek to bridge the gap by drawing on various sources of background knowledge, such as: other ontologies, thesauri, lexical databases, online encyclopedias, and text corpora [11, 14]. These knowledge sources can then be used to implement matching functions that account for spelling variations and synonyms, and that also support some measure of semantic comparison [15].

One approach to measuring semantic similarity of elements is to employ WordNet similarity [16]. However, WordNet offers little coverage of concepts found in real-world ontologies. Another approach is to learn word embeddings directly from text corpora. Word embeddings are distributed word representations that are trained through deep neural networks. Each dimension of the embeddings represents a latent feature of the word, often capturing useful syntactic and semantic properties [17].

Word embeddings have proved to be useful at improving the performance of a wide range of Natural Language Processing (NLP) tasks [18]. Zhang et al. [15] showed that word embeddings learned over Wikipedia can improve the effectiveness of matcher ensembles applied to OAEI benchmark, conference track, and real-world ontologies.

Our own work is similar to that of Zhang et al. [15] but is differentiated in two primary ways. First, we learn word embeddings from a corpus of annotated scientific literature related to the ontologies to be aligned, rather than from Wikipedia. Second, we employ ensemble learning to integrate open source ontology matchers with our word embedding based matcher.

3 Experimental Setup

Our research centers on the hypothesis that the information gleaned from the word embeddings learned from a relevant, annotated corpus would improve matching results within a learned ensemble of existing open source ontology matchers. We tested this hypothesis with systematic experiments using the datasets and techniques described in the following.

3.1 Datasets

To evaluate our ensemble matching system, we used two ontologies of disease vocabularies: the subset of the Online Mendelian Inheritance in Man (OMIM) disease vocabulary, a flat list of disease terms covering genetic disorders; and the ‘Diseases’ branch of the National Library of Medicine’s Medical Subject Headings (MeSH). A third vocabulary, the Comparative Toxicogenomics Database’s (CTD) ‘merged disease vocabulary’ (MEDIC) [19] serves as a reference alignment between OMIM and MeSH. We chose these datasets primarily because there exists a corpus of PubMed titles and abstracts where disease mentions are annotated with the corresponding MEDIC identifiers—such a corpus is needed to train the model from which we train the underlying neural network for our word embedding matcher. In particular, PubTator (a Web-based tool for accelerating manual literature curation) provides an archive of the computer

annotation results for the entire collection of PubMed articles in PubTator[1]. This computer-annotated corpus is generated using the DNorm tool for disease named entity recognition [20].

The data files for our ontologies were collected at the end of 2015 for the MeSH, OMIM, and MEDIC disease vocabularies. The ontology for the MeSH 'Diseases' branch includes 11,344 concepts. The ontology of OMIM genetic disorders includes 8,064 concepts. The MEDIC reference alignment identifies 3,435 direct mappings between MeSH and OMIM concepts. Lastly, the entire PubTator corpus contains 14,412,044 documents.

3.2 Word Embedding Matcher (Word2vec)

Our word embedding matcher uses the similarity scores, as learned by the Word2vec component of the Deeplearning4j library [21], as the confidence for a match between a given pair of ontology concepts. Word2vec is a two-layer neural net that processes text, taking a text corpus as input and outputting a set of feature vectors for words in the corpus. The vectors used to represent words are called *neural word embeddings* and represent a word with numbers based on other neighboring words within the input corpus (see **Table 1**). Given a large enough corpus, Word2vec can make highly accurate guesses about a particular word's meaning—without human intervention—based solely on numerical representations of word features, such as the context of individual words. Word embedding similarity scores are calculated as the cosine similarity of the vectors for a pair of concepts in the MeSH and OMIM ontologies.

Table 1. Examples of neural word embedding vectors learned from the PubTator corpus.

bone	*marrow, (bmt), solid-organ, disseminated, allogeneic, …*
blood	*pressure, rate, hypotension, arterial, concentration, …*
heart	*rate, cardiac, re-infarction, pressure, o2, arterial, …*
liver	*renal, hepatic, failure, acute, function, chronic, …*

Before training the Word2vec model, we preprocess the PubTator corpus so that the annotated phrases for each PubMed document (title and abstract) are replaced by a unique single-token identifier for the corresponding MeSH or OMIM concept. This is necessary because Word2vec learns similarity vectors based on individual words/tokens (and not multi-word phrases). The unique identifier allows us to look up similarity scores for a given pair of concepts from the trained word embedding model. We used Deeplearning4j's suggested configuration: a word window size of 10 for calculating within-sentence word context and the skip-gram technique for predicting the target context, which produces more accurate results on large datasets.

[1] https://ncbi.nlm.nih.gov/CBBresearch/Lu/Demo/PubTator/tutorial/index.html#DownloadFTP

Training the Word2vec model for more than 14 million documents is very time consuming (on the order of weeks). Once the model is built, however, extracting the similarity score for a given pair of terms is fast. The training time can be reduced by distributing the processing with, for example, an Apache Spark cluster.

3.3 Ontology Matching Systems

In addition to the word embedding matcher, we also utilized a number of publicly available ontology matching systems. These matchers are used both alone and as part of a learned ensemble to evaluate the relative impact of the addition of our word embedding matcher. These systems have all participated in past Ontology Alignment Evaluation Initiative (OAEI) campaigns.

LogMap. LogMap [22] is a scalable ontology matching system that utilizes highly optimized data structures to index the input ontologies (both lexically and structurally) to compute an initial set of anchor mappings with corresponding confidence values. These anchors are then used in an iterative process of mapping repair and mapping discovery to uncover new mappings.

AgreementMakerLight (AML). AML is an ontology matching framework based on AgreementMaker [23], one of the leading ontology matching systems. However, whereas AgreementMaker is memory-intensive and was not designed to match ontologies with more than a few thousand concepts, AML is a lightweight system developed with a focus on computational efficiency and is specialized on the biomedical domain but applicable to any ontologies.

Generic Ontology Matching and Mapping Management (GOMMA). GOMMA provides a comprehensive and scalable infrastructure to manage large life science ontologies, but as a generic tool it can be used to match ontologies from other domains [24]. GOMMA preprocesses all information relevant for matching ontology concepts (e.g., name, synonyms, comments) and uses maximal string similarity to generate matches before aggregating the mappings, filtering out any mappings below a certain threshold, and applying constraints to improve the consistency of mappings.

(not) Yet Another Matcher (YAM++). The underlying idea of the YAM++ system is that the complexity and, therefore, the cost of the ontology matching algorithms can be reduced by using indexing data structures to avoid exhaustive pair-wise comparisons [25]. YAM++ preprocesses the input ontologies to calculate the information content of each word to determine the weights of labels. Candidate mappings are passed to a process that uses machine learning to combine several different string-based comparisons to compare the labels/synonyms of entities. Those results are then passed to a structural matcher, which looks at related entities to find more mappings, before combining and filtering the results.

Falcon-AO. Falcon-AO is a prominent component of the Falcon infrastructure for Semantic Web applications [26]. For our datasets, Falcon-AO primarily uses partition-based block matching (PBM), which first divides each ontology into blocks that have a high degree of cohesiveness; then, mappings are discovered by matching similar blocks. The similarity between blocks is a function of the number of "anchors" (alignments with high similarity based on string comparison techniques) that they share.

3.4 Ensemble Learning

We utilize machine learning techniques to determine the weights and confidence level thresholds for each ensemble configuration, allowing for the systematic learning of rules for estimating the correctness of a correspondence based on the output of the different techniques. Our experiments were conducted with the Weka Toolkit [27], using the Weka implementation of the REPTree classifier, a fast decision tree learner which builds a tree using information gain as the splitting criterion and then prunes it using reduced error pruning. Our feature vectors comprise the individual mapping confidence scores for each technique being evaluated as well as a single meta-level feature—average matcher confidence. The inclusion of this meta-level feature is based on the findings of Eckert et al. [12] in which it was found that the most significant feature was not the confidence scores themselves, but the fraction of matchers that found a correspondence. All experiments were conducted with the default Weka classifier settings, making our experiments more easily reproducible.

Dealing with imbalanced data. Each individual matcher can generate mappings with a range of confidence scores between 0.0 and 1.0 and, unsurprisingly, a large number of incorrect mappings appear at low confidence levels. This introduces a problem during classifier training known as *class imbalance*—a large difference in the number of positive and negative instances used to train a classifier (i.e., correct vs. incorrect mappings), which may result in a classifier that is biased towards this majority class. At the extreme, this can lead to a classifier with high accuracy that has actually learned to *always* choose the majority class (i.e., that the mapping is incorrect). In order to account for this when training the classifier, we use a common resampling approach in which the training instance are sampled to provide an even distribution of correct and incorrect training instances. We achieve this by using the Resample filter of the Weka framework for sampling without replacement, and biasing towards a uniform class distribution (i.e., an even split between positive and negative instances).

4 Results

Here we describe the results of our experiments to evaluate the performance of our Word2vec-based word embedding matcher. We analyze the performance of the word embedding matcher both in isolation and by measuring its contribution when combined with one or more existing ontology matching systems, showing that this novel technique adds value that is not identified by standard ontology matching systems.

For the evaluation of each particular classifier configuration, we follow a technique meant to mimic a practical training process for each classifier within the context of scientific literature curation. More specifically, we limit the training of each classifier to a small subset of the mappings produced by the corresponding matchers. We split the training collection into *n* folds, with each fold consisting of approximately 362 instances, and train a separate classifier on each of the *n* individual folds. This is meant to simulate the process of training the classifier with a small number of manually reviewed mappings. 362 was chosen as the approximate fixed size for each fold so that the smallest training collection (YAM++ by itself; 3,628 mappings) would have 10 folds for training. Every evaluation uses the same test collection, consisting of the union of all of the potential mappings generated by each of the matching systems (including Word2vec). This allows for a more accurate comparison of the evaluation results across different classifier configurations. We report the average and standard deviation of the traditional precision, recall, and F-measure metrics across each of the *n* folds for each classifier configuration.

4.1 Word Embedding Similarity Scores

We first analyzed the similarity scores produced by the Word2vec technique, which are the cosine similarity of the vectors for each pair of concepts in the MeSH and OMIM ontologies. For comparison, we built two word embedding models for the PubTator corpus: one with the standard configuration and one providing a list of stop words, which Word2vec ignores during training. The chart in **Fig. 1** shows the raw counts of the correct and incorrect mappings for both of these models.

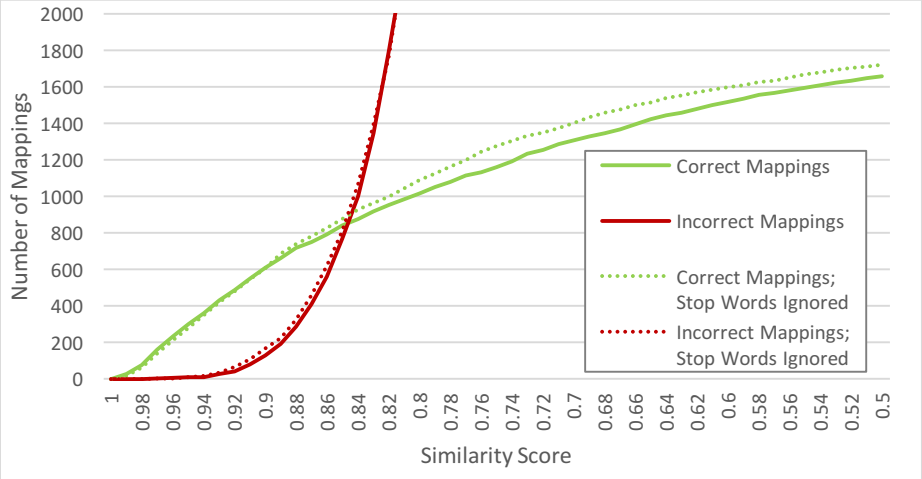


Fig. 1. The raw number of correct and incorrect mappings by Word2vec similarity score for two word embedding models, trained with and without stop words ignored.

The results from both models are very similar, with the global distribution of similarity scores (both correct and incorrect) following a normal distribution. The

Word2vec model that ignores stop words finds slightly more correct mappings when at lower values for the similarity score threshold (i.e., below 0.9). It is understandable that ignoring stop words makes little difference if the window size is sufficient, since the Word2vec model automatically accounts for the information gain afforded by specific context words (which should be near zero for stop words). In both models, the number of incorrect mappings increases drastically as the similarity score threshold decreases, with the number of correct and incorrect mappings being roughly equal with a similarity score threshold of 0.85.

For our experiments, we use similarity scores of at least 0.69. This threshold was chosen so that the number of mappings would be at least twice the size of the larger of the two ontologies (the MeSH ontology contains 11,344 concepts) because a concept in the MeSH ontology may map to more than one concept in the smaller OMIM ontology (8,064 concepts), but not the other way around. By comparison, the number of potential mappings generated by the other ontology matching systems ranges from 3,628 to 7,145. Classifiers trained from the Word2vec similarity scores alone do not perform particularly well (**Table 2**). Surprisingly, precision was high and recall was low, which is the reverse of what we had expected. For our remaining reported experimental results, we use the model with stop words ignored, representing 25,610 total instances (5.6% of which are correct mappings).

Table 2. The average and standard deviation of the F-measure and corresponding precision and recall statistics for each Word2vec word embedding model alone.

	Precision	F-measure	Recall
Word2vec	0.623 ±0.278	0.281 ±0.111	0.190 ±0.082
Word2vec; Stop Words Ignored	0.618 ±0.234	0.301 ±0.099	0.208 ±0.078

4.2 Ensemble Comparisons

For our baseline, we first look at each ontology matching system alone, using our ensemble approach to learn how to distinguish correct from incorrect mappings using only the confidence scores produced by each system (**Table 3**).

The scores for each individual matching system vary widely, which is not particularly surprising given the relatively small fixed-size folds that are used for training each classifier. In the individual configuration, GOMMA and Falcon-AO perform the best on these datasets, with F-measures of 0.590 and 0.546, respectively. Having identified the baseline values for each ontology matching system, we then included the similarity scores generated from our Word2vec word embedding matcher when training a new ensemble for each of the individual ontology matching systems (**Table 3**).

When including the Word2vec similarity scores, we see improved F-measure scores across the board and, in general, the standard deviation for each statistic decreases. The most significant gains are to the recall of the LogMap and AML systems as well as in the precision of LogMap and YAM++. Interestingly, the recall for YAM++ drops when adding Word2vec similarity scores.

Table 3. The average and standard deviation of the F-measure and corresponding precision and recall statistics for each ontology matching system alone and the difference when combined with the Word2vec word embedding matcher.

	Precision	F-measure	Recall
LogMap	0.304 ±0.270	0.260 ±0.269	0.293 ±0.345
Δ LogMap with Word2vec	**+0.243 ±0.179**	**+0.344 ±0.121**	**+0.477 ±0.226**
AML	0.471 ±0.200	0.436 ±0.165	0.530 ±0.148
Δ AML with Word2vec	**+0.131 ±0.123**	**+0.203 ±0.038**	**+0.217 ±0.159**
GOMMA	0.460 ±0.158	0.590 ±0.202	0.821 ±0.282
Δ GOMMA with Word2vec	+0.084 ±0.172	+0.038 ±0.124	+0.025 ±0.239
Falcon-AO	0.500 ±0.122	0.546 ±0.113	0.658 ±0.087
Δ Falcon-AO with Word2vec	+0.039 ±0.179	+0.025 ±0.142	+0.023 ±0.217
YAM++	0.340 ±0.242	0.331 ±0.158	0.705 ±0.288
Δ YAM++ with Word2vec	**+0.236 ±0.106**	**+0.249 ±0.083**	-0.084 ±0.145

Finally, we combined all of the ontology matching systems together to compare the results both with and without Word2vec, as shown in **Table 4**. The F-measure for the model trained using the results from all of the ontology matching systems (without Word2vec) improves over the classifiers trained on the results of each system alone (even if the improvement is only marginal, as in the case of GOMMA). The only evaluation statistics to decrease in the full ensemble configuration are the recall for GOMMA and for YAM++.

Table 4. The average and standard deviation of the F-measure and corresponding precision and recall statistics for all of the ontology matching systems combined and when combined with the Word2vec word embedding matcher.

	Precision	F-measure	Recall
ALL without Word2vec	0.593 ±0.023	0.593 ±0.061	0.683 ±0.165
Δ ALL with Word2vec	+0.053 ±0.151	+0.040 ±0.082	+0.083 ±0.213

Word2Vec contributes value beyond the traditional matching systems: including the Word2vec similarity scores when training the ensemble model boosts recall, precision, and F-measure (the standard deviation across each training fold also increases).

Interestingly, when comparing the performance of the full ensemble classifier (with Word2vec) against the individual matchers each paired with Word2vec, we see that the F-measure for both AML and GOMMA does not change significantly when including the other systems. This would seem to indicate that neither GOMMA nor AML, when combined with Word2vec, are further improved by adding any of the additional matching systems. However, note that GOMMA produces the highest recall of any combination evaluated (0.846 ±0.239), whereas the full ensemble and AML (each including Word2vec) appear to be more balanced as illustrated by their lower recall and higher precision scores.

5 Conclusions and Future Work

In this paper, we have described an ensemble learning approach that augments a collection of ontology matching systems with word embeddings generated from an annotated corpus of relevant scientific literature. We have shown that, within this ensemble approach to ontology matching, the information within word embeddings does contribute to learning an improved model for identifying correct alignments between two ontologies, beyond what state-of-the-art ontology matching systems identify—both individually and in combination. More specifically, the best overall performance (by F-measure) was found in the combination of word embedding-derived similarity scores with either the full ensemble containing *all* of the matching systems under evaluation or the individual AML and GOMMA matching system. However, each of those configurations differed in precision and recall and, therefore, the needs of any particular use case will inform the best configuration for each individual situation.

There are also several items that remain to be answered by future work as well as by our own ongoing research. First, we are currently analyzing the PubTator corpus to extract a list of *multi-word expressions*—using a novel technique for extracting salient variable-length phrases from large text corpora [28]—which we will use in a similar approach to preprocess the corpus and, prior to training the word embedding model, remove all text that is not among the top expressions in the corpus. We also see opportunities to improve upon our ensemble learning approach by providing additional meta-level features when training our ensemble model, such as binary matcher voting, global ontology features, and concept-specific lexical features used by Eckert et al. [12].

Repeating our experiments with different ontologies and/or in a different domain would help to corroborate our results. Training the relevant Word2vec model, however, requires identifying a sufficiently large domain-relevant corpus that is also annotated with concepts from those ontologies. Given a domain-relevant corpus, it may be possible to use an automated system to automatically detect and annotate concept labels in text, as was done by the DNorm disease tagger for the PubTator corpus.

There is also an opportunity to significantly reduce the processing time needed to train a Word2vec model from a given corpus. We briefly explored using Deeplearning4j's support for the Apache Spark cluster-computing framework, but we were unable to fully implement the functionality due to time limitations. With Spark, Deeplearning4j can distribute the processing and train models in parallel for individual shards of the large corpus before iteratively averaging the parameters into a central model.

Lastly, in specific regard to manual biocuration and systematic review processes, we see an opportunity to exploit additional sources of evidence beyond the resulting annotated corpus. More specifically, it may be possible to collect incremental pieces of feedback from work-centered interfaces over the course of a user's normal interaction during biocuration and annotation tasks—for example, while searching for or disambiguating specific concepts for annotating a particular text mention or reference—that can be utilized to further improve ontology matching processes.

Acknowledgments

This work is supported by the US Army Medical Research and Materiel Command under Contract No. W81XWH-13-C-0036.

The views, opinions and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy or decision unless so designated by other documentation.

References

1. The Uniprot Consortium (2015) UniProt: a hub for protein information. Nucleic Acids Res. 43: D204–D212. doi: 10.1093/nar/gku989
2. Sayers EW, Barrett T, Benson DA, Bolton E, Bryant SH, Canese K, Feolo M (2012) Database resources of the national center for biotechnology information. Nucleic acids research 40(D1): D13–D25. doi: 10.1093/nar/gks1189
3. Shimoyama M, De Pons J, Hayman GT, Laulederkind SJ, Liu W, Nigam R, Petri V, Smith JR, Tutaj M, Wang SJ, Worthey E, Dwinell M, Jacob H (2015) The rat genome database 2015: genomic, phenotypic and environmental variations and disease. Nucleic Acids Res. 43: D743–50. doi: 10.1093/nar/gku1026
4. Ghiasvand O, Shimoyama M (2016) Introducing a text annotation tool (OnToMate); assisting curation at rat genome database. In: Proceedings of the 7th ACM international conference on bioinformatics, computational biology, and health informatics (BCB '16). ACM, New York, pp 465-465
5. Poux S, Arighi CN, Magrane M, Bateman A, Wei CH, Lu Z, Boutet E, Bye-A-Jee H, Famiglietti ML, Roechert B (2017) On expert curation and scalability: UniProtKB/Swiss-Prot as a case study. Bioinformatics btx439. doi: https://doi.org/10.1093/bioinformatics/btx439
6. Rodriguez-Esteban R (2015) Biocuration with insufficient resources and fixed timelines. Database: The Journal of Biological Databases and Curation 2015; 2015: bav116. doi:10.1093/database/bav116
7. Marshall C, Brereton P (2015) Systematic review toolbox: A catalogue of tools to support systematic reviews. In: Proceedings of the 19th international conference on evaluation and assessment in software engineering. ACM, New York, p 23
8. Choong MK, Galgani F, Dunn AG, Tsafnat G (2014) Automatic evidence retrieval for systematic reviews. J Med Internet Res 2014;16(10): e223. doi: 10.2196/jmir.3369
9. Wallace BC, Kuiper J, Sharma A, Zhu MB, Marshall IJ (2016) Extracting PICO sentences from clinical trial reports using supervised distant supervision. Journal of Machine Learning Research, 17(132): 1–25
10. Basu T, Kumar S, Kalyan A, Jayaswal P, Goyal P, Pettifer S, Jonnalagadda S (2016) Systematic reviews by automatically building information extraction training corpora. arXiv preprint arXiv:1606.06424.
11. Shvaiko P, Euzenat J (2013) Ontology matching: state of the art and future challenges. IEEE transactions on knowledge and data engineering 25(1): pp 158–176. doi: 10.1109/TKDE.2011.253
12. Eckert K, Meilicke C, Stuckenschmidt H (2009) Improving ontology matching using meta-level learning. In: Aroyo L, et al. (eds). LNCS, volume 5554. Springer International Publishing, Cham, Switzerland, pp 158-172. doi: https://doi.org/10.1007/978-3-642-02121-3
13. Gal A (2011) Uncertain schema matching. Synthesis Lectures on Data Management 3(1):1–97

14. Faria D, Pesquita C, Santos E, Cruz IF, Couto FM (2014) Automatic background knowledge selection for matching biomedical ontologies. PLoS ONE 9(11): e111226. doi: https://doi.org/10.1371/journal.pone.0111226
15. Zhang Y, Wang X, Lai S, He S, Liu K, Zhao J, Lv X (2014) Ontology matching with word embeddings. In: Maosong S, Liu Y, Zhao J (eds) Chinese computational linguistics and natural language processing based on naturally annotated big data. Springer International Publishing, Cham, Switzerland, pp 34–45. doi: https://doi.org/10.1007/978-3-319-12277-9
16. Lin F, Sandkuhl K (2008) A survey of exploiting wordnet in ontology matching. In: Bramer M (ed) Artificial intelligence in theory and practice, vol 2. Springer US, New York, pp 341–350. doi: 10.1007/978-0-387-34747-9
17. Turian J, Ratinov L, Bengio Y (2010) Word representations: a simple and general method for semi-supervised learning. In: Proceedings of the 48th annual meeting of the association for computational linguistics. Association for Computational Linguistics, Stroudsburg, PA, pp 384–394
18. Li Y, Yang T (2018) Word embedding for understanding natural language: survey. In: Srinivasan S. (ed) Guide to big data applications. Studies in big data, vol 26. Springer International Publishing, Cham, Switzerland, pp 83–104. doi: 10.1007/978-3-319-53817-4
19. Davis AP, Wiegers TC, Rosenstein MC, Mattingly CJ (2012) MEDIC: a practical disease vocabulary used at the comparative toxicogenomics database. Database: The Journal of Biological Databases and Curation 2012; 2012: bar065. doi: 10.1093/database/bar065
20. Leaman R, Islamaj Doğan R, Lu, Z (2013). DNorm: disease name normalization with pairwise learning to rank. Bioinformatics 29(22): 2909–2917
21. Deeplearning4j Development Team (2016) Deeplearning4j: Open-source distributed deep learning for the JVM. https://deeplearning4j.org/about. Accessed 27 July 2017
22. Jiménez-Ruiz E, Cuenca Grau B (2011) LogMap: Logic-based and scalable ontology matching. In: Aroyo L et al. (eds) The semantic web – ISWC 2011. ISWC 2011. Lecture Notes in Computer Science, vol 7031. Springer, Berlin, Heidelberg, pp 273–288. doi: https://doi.org/10.1007/978-3-642-25073-6_18
23. Faria D, Pesquita C, Santos E, Palmonari M, Cruz IF, Couto FM (2013) The AgreementMakerLight ontology matching system. In: Meersman R et al. (eds) On the move to meaningful internet systems: OTM 2013 Conferences. OTM 2013. Lecture Notes in Computer Science, vol 8185. Springer, Berlin, Heidelberg, pp. 527–541. doi: https://doi.org/10.1007/978-3-642-41030-7_38
24. Kirsten T, Gross A, Hartung M, Rahm E (2011) GOMMA: a component-based infrastructure for managing and analyzing life science ontologies and their evolution. Journal of Biomedical Semantics 2(1): 6. doi: 10.1186/2041-1480-2-6
25. Duyhoa N, Bellahsene Z (2014) Overview of YAM++-(not) Yet Another Matcher for ontology alignment task. Dissertation, LIRMM
26. Hu W, Qu Y (2008) Falcon-AO: A practical ontology matching system. Web Semantics: Science, Services and Agents on the World Wide Web 6(3): 237–239. doi: 10.1016/j.websem.2008.02.006
27. Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, Witten I (2009) The WEKA data mining software: an update. ACM SIGKDD Explorations Newsletter 11(1): 10–18. doi: 10.1145/1656274.1656278
28. Shang J, Liu J, Jiang M, Ren X, Voss CR, Han J (2017) Automated phrase mining from massive text corpora. *arXiv preprint arXiv:1702.04457*

Ontology Matching for Patent Classification

Christoph Quix[1,2], Sandra Geisler[2], Rihan Hai[1], Sanchit Alekh[1]

[1] Databases and Information Systems, RWTH Aachen University, Germany
[2] Fraunhofer-Institute for Applied Information Technology FIT, Germany
[1]lastname@dbis.rwth-aachen.de
[2]firstname.lastname@fit.fraunhofer.de

Abstract. Interdisciplinary research and development projects in medical engineering benefit from well selected collaboration partners. The process of finding such partners from often unfamiliar fields is difficult, but can be supported by an expert profile that is based on patent analysis and classifying the patents to competence fields in medical engineering. Patent analysis and categorization are difficult and require the analysis of the semantic content. Hence, we propose a twofold approach using a large controlled vocabulary, a smaller competence field ontology, and an alignment between them to assign patents to a certain competence field. The approach has two parts: a *Topic Map* approach and a *Publication approach*. We evaluate these approaches and its components in several ways. Furthermore, we compare four different ways to assign a patent to a competence field and show that the semantic wealth of a large biomedical ontology is beneficial to the classification task.

1 Introduction

Ontology matching has been an active research area for more than 10 years [17,18]. Ontologies are used to describe a domain of interest by concepts and relationships between them, and to provide a formal description of these relationships. Thus, although the aim of ontology matching seems to be the matching of classes and properties, usually its actual intention is *to match elements of the domains described by the ontologies*. An example for such a 'domain matching' task is patent classification in which patents should be assigned to a class in a classification [4].

While a classification scheme or taxonomy can be easily represented as an ontology, representing the content of a patent as an ontology or describing the patent with elements of an ontology is more challenging. Patents have their own specific language and use a terminology that is different from a typical research publication. Patents are classified using the International Patent Classification (IPC) system; however, this is too general for a detailed patent analysis [12]. On the other hand, patent data is available in a structured form (usually XML) from patent offices, which simplifies the pre-processing and extraction of basic information such as title, abstract, and authors. Furthermore, they are often also available in multiple languages; at least, the bibliographic information and

abstract is available in English, which solves the problem of multi-lingual documents.

We are aiming at building a recommender system for research projects in medical engineering (ME) [7] in the context of the mi-Mappa project[3]. In ME researchers from several disciplines (e.g., biology, medicine, mechanical engineering, computer science) work jointly on a research project. Furthermore, ME is a highly innovative domain with short product cycles requiring a fast translation of research results into applicable products [2]. While on the one hand, a publication list of a researcher provides a good basis for creating an author profile [14], on the other hand a list of patents allows to characterize the ability of a researcher to develop inventions and market-ready products. Hence, we concentrate mainly on the analysis of patents.

To address the problem of patent terminology, we exploit explicit references to scientific publications and their semantic annotations. In ME, most of the publications appear in journals or conferences that are indexed by PubMed[4]. PubMed uses MeSH[5], a rich controlled vocabulary with a hierarchical structure, to annotate the publications. Thus, to retrieve a MeSH annotation for a patent, we lookup the references to research articles in PubMed and retrieve the corresponding MeSH terms.

Using references to scientific publications is only one aspect in our approach for patent classification. The overall approach, depicted in Figure 1 consists of two complementary sub-approaches: the *Topic Map Approach* (TMA) and the *Publications Approach* (PBA). Both approaches utilize two ontologies - a competence field (CF) ontology and an ontology with comprehensive medical knowledge (MeSH) - and an alignment between them.

For the Publication Approach, excerpts of publication databases, as well as their associated MeSH terms are imported into our Data Lake (DL) system Constance [8]. The data lake can then be queried on-the-fly for publications cited by the currently processed patent, as well as the MeSH terms that are pertinent to each of these publications. For the categorization of the input patent with the TMA, the topic with the highest probability in the topic map (or multiple topics if they have the same probability) is retrieved. Each term characterizing the topic is compared with all concepts in the MeSH ontology resulting in a set of matching concepts.

Thus, for both approaches, we have a list of related concepts from the MeSH ontology. To establish a link to the competence field ontology, which we have created to describe the innovation areas in medical engineering (see section 2), we use ontology matching.

There are several questions arising when we analyze the presented approach. Creating an alignment between ontologies and the use of a huge medical ontology in this context require a high amount of resources in terms of memory and CPU power. Hence, we need to know if the effort using it is worth it. Furthermore,

[3] `http://www.dbis.rwth-aachen.de/mi-Mappa`
[4] `https://www.ncbi.nlm.nih.gov/pubmed/`
[5] Medical Subject Headings, `https://www.nlm.nih.gov/mesh/`

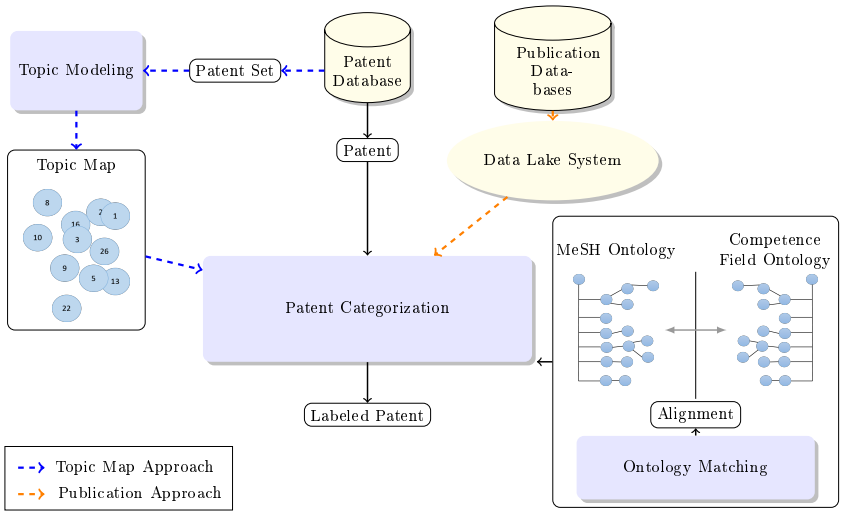


Fig. 1. The Overall Architecture

it is of interest if the quality and size of the alignment between the ontologies have an impact on the results. A special problem is to rate the quality of the alignment without a reference alignment. To answer these questions we present the following contributions in this paper:

- We analyze and select medical ontologies to use them as a basis for the creation of the CF ontology and as a single point of entry to identify the semantics of patents and publications.
- We describe the process of designing the competence field ontology and rate its quality based on approved methodologies.
- We create different alignments between the CF ontology and the medical ontology with different matcher configurations and compare their quality.
- We compare the results of four different approaches to categorize a patent: (1) Topic Map Approach with direct comparison of terms with concepts of the CF ontology (i.e., using no ontology matching techniques), (2) Publication Approach, (3) Topic Map Approach, (4) combination of Topic Map Approach and Publication Approach. Approach (2) and (3) use the alignment computed by ontology matching.

The rest of this paper is structured as follows. In Section 2 we explain the design of the CF ontology. Furthermore, the selection process of the utilized medical ontologies is explained (first results about these issues were reported in [7]). In Section 3 we describe the approaches to establish a link between patents and competence fields. In Section 4 the four approaches to categorize patents into competence fields will be evaluated. Finally, we discuss related work in Section 5 and conclude the paper in Section 6.

2 Modeling and Selection of Ontologies

Our assumption is that a huge medical ontology (or a set of them) and mappings to a smaller competence field ontology (CFO) will help to more easily classify patents into competence fields. The idea is somehow similar to a smart multi-level filter. First we retrieve terms describing the content of a patent (either from the topic map or the cited publications). These terms are compared to concept names in a huge medical taxonomy using string similarity measures. The most similar ones are selected, which results in a potentially long list of concepts. Afterwards we filter further and search for mappings from the concepts and their predecessors to concepts of the smaller competence field ontology using more intelligent matchers. This leads to scores which identify the membership confidence to the competence fields.

To implement this approach two foremost things have to be done: (1) we have to model the competence field ontology and (2) we need to evaluate and select comprehensive medical ontologies. For the design of ontologies there exist several acknowledged methodologies, such as METHONTOLOGY [6], TOVE, or the work by Noy and McGuinness [13]. The NeOn methodology [19] is a more recent approach which combines ideas of the former methods. The methodology describes nine scenarios for building ontologies and ontology networks [19].

To create the CFO, we started from the descriptions in [15,3] and also used an extended description of ME domain experts. As the six competence fields are the categories we want to assign to the patents, we use these (and only these) as first level concepts in the ontology. All further concepts will be subconcepts of these. This approach corresponds to the *reusing and reengineering non-ontological resources* scenario of the NeOn methodology. To find subconcepts, we had analyzed the detailed description of the CFs by the domain experts. Firstly, we extracted a preliminary selection of 174 terms which we used to make a first draft of a preliminary ontology on which domain experts commented using a custom web front end for the review of ontologies.

In parallel we searched for one or multiple large biomedical taxonomies. We need these taxonomies for two things. First, we want to extend the basic CFO we created before with more terms to describe the competence fields in more detail. Second, we need the large ontology as entry point to find terms describing the patents and with the alignment to the CFO we can determine the corresponding competence fields. This corresponds to the sixth scenario of the NeOn methodology, namely *reusing, merging and reengineering ontological resources*. The first step in this scenario is the *ontological resource reuse process*, starting with the *Ontology Search* [19]. Hence, we searched for ontologies with domain specific search engines as described in [7]. We used the Bioportal[6] search engine, the Ontology Lookup Service[7], and the Ontobee[8] search engine using the preliminary list of terms to have a broad overview. Afterwards we carried out

[6] http://bioportal.bioontology.org
[7] http://www.ebi.ac.uk/ontology-lookup
[8] http://www.ontobee.org

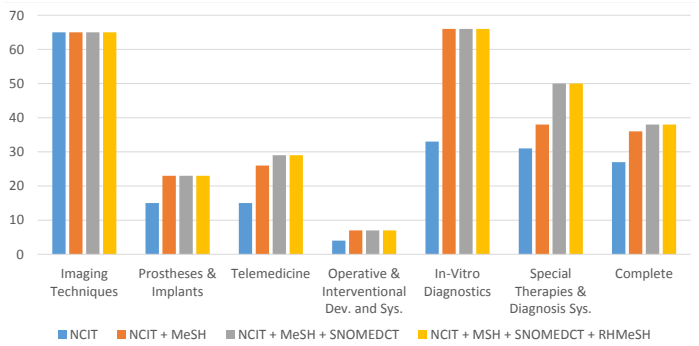


Fig. 2. Coverage based on Combination of Ontologies

the *Ontology Assessment and Comparison* steps [19]. The most promising four ontologies found are the National Cancer Institute (NCIT) Thesaurus, the Systematized Nomenclature of Medicine - Clinical Terms (SNOMEDCT), MeSH, and the Robert Hoehndorf Version of MeSH (RHMeSH). To identify if they satisfy our needs, we did a coverage analysis, where the coverage is the percentage of the competence field terms present in each of the ontologies. No single ontology covered all competence fields to a satisfying degree; some reached more than 60% for one competence field but only about 20% for the other fields (e.g., NCIT covers 'Imaging Techniques' well, but not the other fields).

Hence, we decided to analyze the coverage by adding one ontology after another to see the gain of adding further ontologies. We used the most promising ontologies identified before and started with the NCI Thesaurus. Figure 2 shows the results.

It can be noted, that we gain about 10% coverage using all ontologies. The biggest gain is achieved by adding the MeSH ontology. Thus, we decided to use the NCIT and the MeSH ontologies to extend the CFO, as this was a good compromise between coverage and complexity. For the matching of the biomedical ontology to the CFO we first picked only one ontology to keep the computational overhead during runtime low. If it does not give us satisfying results, we will add more ontologies and also align them with the CFO. One possibility would be also to use the UMLS which is a superset of many medical ontologies, but it is really large, which could lead to performance problems. For now, we selected the Robert Hoehndorf MeSH[9] as it has a good coverage and is available in the OWL format.

The next steps to develop the CFO are the *ontology aligning and ontology merging* step and the *ontological resource engineering process*. We proceeded in these steps as follows. We took the extracted terms, the so far found concepts from the coverage analysis, and the detailed description of the innovation fields, and carried out an extended search in the MeSH Browser[10] and the NCIT Brow-

[9] `https://bioportal.bioontology.org/ontologies/RH-MESH`
[10] `https://meshb.nlm.nih.gov/search`

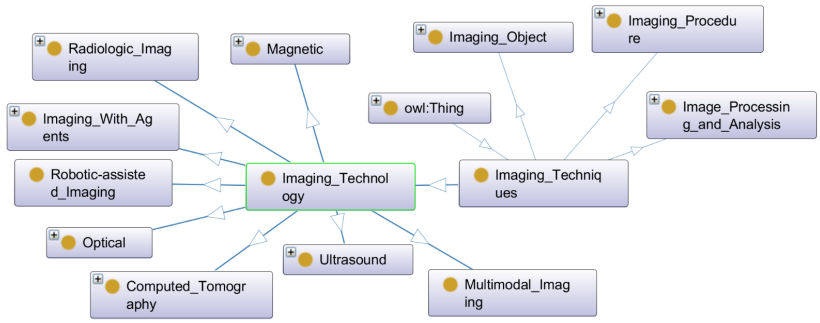


Fig. 3. The Imaging Technique Concept

ser[11] for these and related concepts. We analyzed the hierarchical structure of each of the found concepts and decided for each concept if it is adopted into the CFO. Where applicable we also adopted the inheritance relationship of concepts. We extended and restructured the CFO in cycles, i.e., according to [19] we did a *re-conceptualization* on different levels for the CFO and for the concepts from the biomedical ontologies. For the upper levels of the CFO we designed categories which fit better to our purposes for categorizing terms for medical engineering. We used a mind mapping technique and a bottom-up approach as for example described by Noy and McGuinness [13] to refine the design. As an example, the Imaging Techniques concepts and the concepts of the concept Imaging_Technology (2nd level) are visualized in Figure 3.

The ontology has been implemented in OWL using the NeOn toolkit[12]. We evaluated the CFO also in tests in the complete process of patent categorization. We noticed that the initial results were not satisfying because some competence fields were not represented well in the CFO. Hence, we did a frequency analysis of the MeSH terms from the Publications Approach. We made a ranked list of MeSH concepts based on how often they have been searched for, but did not lead to matches in the CFO. Based on this list we added more useful concepts to the CFO (no trivial, misleading terms, such as *Human*, but for example *Gene Expression Regulation*). The current CFO consists of 529 concepts and can be downloaded at `http://dbis.rwth-aachen.de/cms/projects/mi-mappa/CFO.owl`.

3 Matching of Ontologies and Topic Maps

As explained above, we are using three different basic approaches and one combined approach to classify patents. Figure 4 gives an overview of the different approaches.

#1: TMD (Topic Map with Direct Mapping): In this approach, we match the terms extracted from the topic maps directly with the competence

[11] `https://ncit.nci.nih.gov/ncitbrowser/`
[12] `http://neon-toolkit.org`

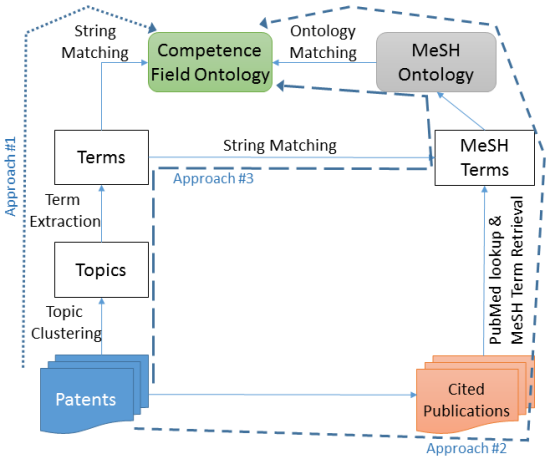


Fig. 4. The different approaches used for Evaluation

field ontology. This can be seen as a base line as it does not use a semantically rich ontology as intermediate component, but only uses string matching to match terms and ontology elements.

#2: PBA (Publication-Based Approach): This approach uses the MeSH terms attached to publications which are referenced by a patent. Then, we use an alignment between the CFO and MeSH to compute a score for the relationship between a patent and a competence field.

#3: TMA (Topic Map Approach): Here, we also use topic mapping (as in approach #1) to create initial clusters of patents and extract terms occurring frequently in these clusters. These terms are then matched with the concepts of the MeSH ontology. Using the same alignment as in the second approach, a relationship to the CFO is established.

#4: COM (Combined Approach of #2 & #3): This is a combination of PBA and TMA, with an emphasis on the results of PBA.

As the approaches TMD and TMA are based on topics, we first briefly explain this part, before we present how we did the alignment between of CFO and MeSH, and describe the publication-based approach.

3.1 Topic Mapping

A basic set of patents is used to build a topic map. Firstly, the corpus of documents is preprocessed (stemming, removing stop words, etc.) and a Document-Term-Matrix (DTM) is created. The matrix is input to a Latent Dirichlet Allocation (LDA) algorithm with the Gibbs sampling algorithm for estimation and variational expectation maximization [11]. The LDA determines a fixed number of topics which are each described by a fixed number of stemmed terms. To each patent in the basic patent set topics are assigned with a probability. The topic map and the assignments are stored in a database.

We evaluated different numbers of topics and different numbers of terms extracted for each topic (e.g., 10, 30, 50, etc.). As computation of the subsequent steps increases with a higher number of topics and terms, we used 50 topics and 50 terms for our evaluation in Section 4. As the TMD approach matches the terms directly with the CFO, no further processing on the extracted terms is done in this case. We just do a similarity calculation using a normalized *Longest Common Subsequence* [10] algorithm. In our tests, we found that a threshold value of 0.5 for the string similarity provides the best compromise.

For the categorization of the input patent with the TMA, the topic with the highest probability in the topic map (or multiple topics if they have the same probability) is retrieved. Each term characterizing the topic is compared with all concepts in the medical ontology resulting in a set of matching concepts. For each of the concepts in the set direct mappings and mappings of parent concepts are collected from the alignment and it is determined to which competence field the matching concept in the CF ontology belongs. From the similarities average scores are calculated for each term and each competence field. Based on this, an average score is calculated from all terms for the topic(s) of the patent. Hence, for each patent we have a score for each of the competence fields and normalize these, such that all scores add up to 1.

3.2 Ontology Matching

To rate how strong a patent or publication is related to a certain competence field, we need to match the describing terms either extracted from publications or from the topic map to terms describing the competence fields. In preparation to this step, we create an alignment between the selected MeSH ontology and the CFO. The alignment constitutes of a set of mappings between the concepts of the two ontologies. This means, for each mapping we have a pair of concepts and a similarity value. As we do not try to re-invent the wheel, we used AgreementMakerLight [5] as it produced constantly good results in the recent OAEI campaigns and also performs well for large biomedical ontologies. AgreementMakerLight is able to combine different matchers to create an alignment. We used the string matcher, the word matcher, the structural matcher, the lexical matcher, the cardinality filter, and the coherence filter. As a similarity threshold we used a value of 0.6. The matchers have been combined in a hierarchical way and the default settings for each matcher have been used.

Currently, we are also testing other settings and their impact on the quality of patent classification results. First experiments show, that slightly relaxed filter settings (e.g., not using a cardinality filter) increases the number of mappings and therefore, also improves the classification result.

3.3 Publication-based Approach

We queried the web service of EPMC[13] to retrieve the metadata of the papers referenced in our patent dataset. To extract the references from the patent

[13] European PubMed Central, `https://europepmc.org/`

data, we use a pattern-based approach similar to the FreeCite citation parser[14]. Luckily, the patent data is semi-structured such that the citations can be clearly identified. Nevertheless, for a large fraction of the patents, we are not able to retrieve MeSH terms from referenced publications (either because the referenced publication does not appear in PubMed or the citation is incorrect).

The retrieved metadata for each referenced publication is then stored in our Data Lake system Constance [8] from which it is accessed during patent processing.

Subsequently, we use a process which is similar to the TMA. In both cases, we have a list of MeSH terms as input. For each of the terms in the list, the mappings are determined as before and average scores per competence field are calculated and normalized for each patent.

3.4 Combined Approach

In the combined approach (COM), if both approaches TMA and PBA deliver results, the results are combined and overall scores for each competence field are determined. In all cases, we assign at most three competence fields to a patent. In most cases, only one competence field is assigned to a patent as the other competence fields do not exceed a certain threshold. Thus, we take the intersection of competence fields computed by TMA and PBA. If this is not empty, we take this result (because both approaches are sure about a result). If the intersection is empty, we take the competence fields with the highest scores from TMA and PBA.

4 Evaluation

In our experimental setup, we compare the aforementioned approaches. For the analysis of patents, we need a comprehensive data basis with high data quality. In the course of the mi-Mappa project, a subset of the PATSTAT database (2016 Spring edition, version 5.07) published by the European Patent Office (EPO) is used. For our purposes, we selected patents issued by a German (DE) or British (UK) authority after 2004, which are from the medical domain (CPC class A61), and which have an English abstract and title. This results in a set of 26,814 patents. For about 4,500 patents of this set, we are able to retrieve MeSH terms for the referenced publications. From this set, we randomly selected 59 patents to do a manual assignment to competence fields to evaluate our approaches. A more extensive expert evaluation is currently being setup. In addition, we plan also to evaluate our approach to the results of our project partners who apply a supervised learning approach using Support Vector Machines [9].

For TMA, we experimented with various configurations for the number of topics and their associated terms. We observe that with a relatively small number of topics and terms, e.g. 10 or 20, the terms are extremely broad-based and do

[14] http://freecite.library.brown.edu/

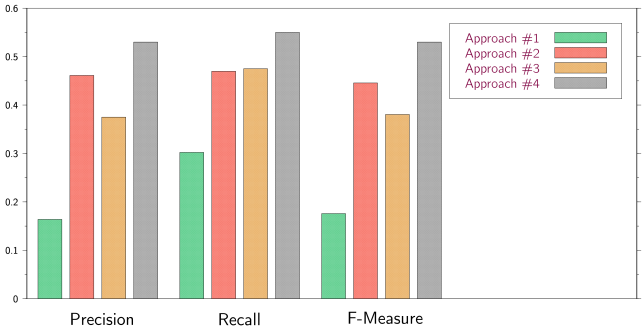


Fig. 5. Comparison of the precision, recall and f-measure for the different approaches

not provide meaningful matches with the MeSH ontology or the CFO. Therefore, based on the results, we chose the number of topics, as well as the number of terms to be 50 for our default test configuration.

Fig. 5 summarizes the findings from our experiments for the aforementioned approaches. It is obvious that all three of our proposed approaches #2, #3, and #4 perform significantly better than the baseline approach #1. All three evaluation parameters, i.e., precision, recall, and the f-measure are worse for the baseline approach. In contrast, when the MeSH ontology is used for matching the ontology terms (*#3*), the precision and f-score are 0.375 and 0.38, respectively, which are more than the doubled values of corresponding values produced by *#1*. The PBA performs even better, resulting in precision, recall, and f-score values of 0.46, 0.47 and 0.44, respectively. However, the combined approach #4 significantly outperforms all the others, and results in precision, recall, and f-measure values of 0.53, 0.55 and 0.53, respectively. Indeed, we found that in the case of the TMD-approach #1, there were a lot of erroneous matches, which led to non-distinctive results for the CFO assignment. These results affirm the superiority of techniques which use a comprehensive biomedical ontology and ontology matching for patent classification tasks.

5 Related Work

There are only few works that apply ontology matching in the context of patent analysis. Semantic similarities (based on ontology matching) and case-based reasoning have been applied in the design of invention processes which use patent analysis to study related works. Patent analysis using ontologies has been applied especially for patent search [1]. A patent search request can be represented as an ontology or as a set of concepts of an existing ontology, which is then matched with the ontologies representing the knowledge of patents [16]. Another example is the PatExpert system which uses a network of ontologies and knowledge bases to enable patent search, classification, and clustering [21]. Trappey et al. propose a system that calculates the conditional probability that, given a specific text chunk is present in the document, the chunk is mapped to a specific concept of a given ontology [20]. Patent similarity is then based on

the number of common matched concepts. This approach restricts the clustering to the terms of the ontology which might lead to missing important terms not present in the ontology.

6 Conclusion

Patent analysis is a complex topic as patents use their own language and terminology. Even for humans used to research publications, patents are difficult to understand. Thus, typical approaches for classifying patents might fail.

In this paper, we investigated an ontology-based approach to assign patents to competence fields in medical engineering. We developed two different approaches and a combined approach that are based on a large biomedical ontology, its alignment to the competence field ontology designed by us, and other ontology matching techniques. We have shown that these more elaborated approaches outperform an approach that directly matches terms of patents with the competence field ontology.

However, the overall f-measure of about 55% for the combined approach is not yet satisfying. One problem is the small set of patents for which we have assigned competence fields that we can use as a ground truth. This will be extended with a larger expert evaluation in which patents will be classified by several experts. Even humans might disagree on the assignment of a patent to a competence field; therefore, we will have multiple expert opinions for one patent. We will also work on fine tuning and optimizing our approach. So far, we focused on the quality of the result, and did not worry too much about the performance. Still, we think that the area of patent classification is an interesting field which could benefit more from the results in ontology matching.

Acknowledgements

This work has been supported by the Klaus Tschira Stiftung gGmbH in the context of the mi-Mappa project (`http://www.dbis.rwth-aachen.de/mi-Mappa/`, project no. 00.263.2015). We thank our project partners from the Institute of Applied Medical Engineering at the Helmholtz Institute of RWTH Aachen University & Hospital, especially Dr. Robert Farkas, for the fruitful discussions of the approach and for providing the patent data.

References

1. D. Bonino, A. Ciaramella, F. Corno. Review of the state-of-the-art in patent information and forthcoming evolutions in intelligent patent informatics. *World Patent Information*, **32**(1):30–38, 2010.
2. BVMed. Branchenbericht Medizintechnologien 2015. www.bvmed.de/branchenbericht, June 2015.
3. Deutsche Gesellschaft für Biomed. Technik im VDE. Empfehlungen zur Verbesserung der Innovationsrahmenbedingungen für Hochtechnologie-Medizin. *Tech. rep.*, VDE, 2012.

4. C. J. Fall, A. Törcsvári, K. Benzineb, G. Karetka. Automated categorization in the international patent classification. In *ACM SIGIR Forum*, pp. 10–25. 2003.
5. D. Faria, C. Pesquita, B. S. Balasubramani, C. Martins, J. Cardoso, H. Curado, F. M. Couto, I. F. Cruz. OAEI 2016 results of AML. In *Proc. 11th Intl. Workshop on Ontology Matching*, pp. 138–145. 2016.
6. M. Fernández-López, A. Gómez-Pérez, N. Juristo. Methontology: from ontological art towards ontological engineering. In *Proc. Symposium on Ontological Engineering of AAAI*. 1997.
7. S. Geisler, R. Hai, C. Quix. An Ontology-based Collaboration Recommender System using Patents. In *Proc. Intl. Conf. on Knowledge Engineering and Ontology Development (KEOD)*, pp. 389–394. Lisbon, Portugal, 2015.
8. R. Hai, S. Geisler, C. Quix. Constance: An Intelligent Data Lake System. In F. Özcan, G. Koutrika, S. Madden (eds.), *Proc. Intl. Conf. on Management of Data (SIGMOD)*, pp. 2097–2100. ACM, San Francisco, CA, USA, 2016.
9. N. Hamadeh, M. Bukowski, T. Schmitz-Rode, R. Farkas. Cooperative Patent Classification as a mean of validation for Support Vector Machine Learning: Case Study in Biomedical Emerging Fields of Technology. In *51. Jahrestagung der Biomedizinischen Technik (BMT)*. 2017.
10. D. S. Hirschberg. Algorithms for the longest common subsequence problem. *Journal of the ACM (JACM)*, **24**(4):664–675, 1977.
11. K. Hornik, B. Grün. topicmodels: An R package for fitting topic models. *Journal of Statistical Software*, **40**(13):1–30, 2011.
12. K.-K. Lai, S.-J. Wu. Using the patent co-citation approach to establish a new patent classification system. *Information Processing & Mgmt.*, **41**(2):313–330, 2005.
13. N. F. Noy, D. L. McGuinness. Ontology Development 101: A Guide to Creating Your First Ontology. *Tutorial*, Stanford University, 2001.
14. J. Portenoy, J. D. West. Visualizing Scholarly Publications and Citations to Enhance Author Profiles. In *Proc. WWW*, pp. 1279–1282. Perth, Australia, 2017.
15. C. Schlötelburg, C. Weiß, P. Hahn, T. Becks, A. C. Mühlbacher. Identifizierung von Innovationshürden in der Medizintechnik. *Tech. rep.*, Bundesministeriums für Bildung und Forschung, October 2008.
16. A. Segev, J. Kantola. Patent Search Decision Support Service. In *7th Intl. Conf. on Information Technology: New Generations (ITNG)*, pp. 568–573. 2010.
17. P. Shvaiko, J. Euzenat. A Survey of Schema-Based Matching Approaches. *Journal on Data Semantics*, **IV**:146–171, 2005. LNCS 3730.
18. P. Shvaiko, J. Euzenat. Ontology Matching: State of the Art and Future Challenges. *IEEE Transactions on Knowledge and Data Engineering*, **25**(1):158–176, 2013.
19. M. C. Suárez-Figueroa. *NeOn Methodology for building ontology networks: specification, scheduling and reuse*. Ph.D. thesis, Univ. Politecnica de Madrid, 2010.
20. A. J. Trappey, C. V. Trappey, F.-C. Hsu, D. W. Hsiao. A fuzzy ontological knowledge document clustering methodology. *IEEE Trans. on Systems, Man, and Cybernetics, Part B*, **39**(3):806–814, 2009.
21. L. Wanner, R. Baeza-Yates, S. Brügmann, J. Codina, B. Diallo, E. Escorsa, M. Giereth, Y. Kompatsiaris, S. Papadopoulos, E. Pianta, et al. Towards content-oriented patent document processing. *World Patent Information*, **30**(1):21–33, 2008.

Extension of the M-Gov Ontology Mapping Framework for Increased Traceability

Anuj Singh, Christophe Debruyne, Rob Brennan, Alan Meehan and Declan O'Sullivan

ADAPT Centre, School of Computer Science and Statistics, Trinity College Dublin, Ireland
{singh.anuj, christophe.debruyne, rob.brennan, alan.meehan, declan.osullivan}@adaptcentre.ie

Abstract. This paper describes an extension to the M-Gov framework that captures queryable metadata about matcher tools that have been utilized, the users involved, and the discussions of the users, during the generation of alignments. This increases the traceability in an alignment creation process and enables an evaluator to more deeply interpret and evaluate an alignment, e.g. for reuse or maintenance. This requires precise information about the alignments being encoded and the decisions undertaken during their creation. This information is not captured by state of the art approaches in a queryable format. The paper also describes an experiment that was undertaken to examine the effectiveness of our approach in enabling the traceability in the alignment creation process. In the experiment, stakeholders created an alignment between two different datasets. The results indicate that the users were 93% accurate while creating the alignment. The major traceability achievements demonstrated for the test groups were 1) level of participation of various users of a group during alignment creation; 2) most discussed correspondences by users of a group; and 3) accuracy of a group in creating alignment.

Keywords: Ontology Matching, Ontology Alignment, Mapping governance

1 Introduction

Ontology mapping is required to overcome the problem of semantic heterogeneity and facilitate interoperability between ontology-based systems that share the same concepts but have the different representation of those concepts [1], [2]. Creation and maintenance of ontology mapping is a difficult task in several aspects [16], one of the aspects, which we focus on this paper is traceability in the alignment creation process.

Alignments are built for a purpose like data integration or a link data mashup for a specific group of stakeholders. Creation of an alignment is a non-trivial task, as it requires these stakeholders to collaborate. In [4], we suggested an approach, which allows stakeholders to collaborate for creating an alignment by using a Mapping Governance framework. An initial implementation of the approach is also outlined in [4], which we now term the M-Gov framework. The framework captures the metadata during alignment creation, which enables the traceability in an alignment creation process.

Traceability in [3] refers to "the ability to follow the life of a requirement in a forward or backward direction". Similarly, the traceability in an alignment creation process will allow one to trace the following for a correspondence: decisions about a correspondence; rationale for the decisions; and the stakeholders who were involved in the decision making process. The approach we introduced in [4] suggested capturing metadata information about the matcher used, the contributors and their discussions during an alignment creation process. Our intuition was that capturing such information would increase traceability in the alignment creation process, as this will not only allow one to formulate queries to look for existing alignments but also to formulate questions such as "which stakeholder participated the most in alignment creation" or "which correspondence was mostly discussed by stakeholders".

In this paper, we first describe how we have extended the M-Gov framework by supporting stakeholders during the Match phase (Section 3). First, the Alignment API 4.8 is used to discover candidate correspondences between two different datasets. Then stakeholders are allowed to discuss each identified correspondence displayed on a web page using a grid table. The paper also describes (Section 4) the initial evaluation that we have undertaken. Specifically, the research question under investigation during our evaluation was to what extent captured metadata allows tracing of: the most discussed correspondences by stakeholders, the level of participation of stakeholders, and the decisions taken by a group of stakeholders for a correspondence?

In summary, the contribution of this paper is as follows: Firstly by extending the M-Gov framework to enable tracebility in an alingment creation process. Secondly, we have provided a detailed description of the alignment creation process. Thirdly we have provided evidence that metadata captured in the M-Gov framework enables traceability in an alignment creation process.

The paper is organized as follows: Section 2 provides an overview of the background information; Section 3 outlines the match phase of the M-Gov framework; Section 4 presents an evaluation of the experiment that was undertaken; Section 5 sheds some light on the related work; and conclusions are drawn in section 6.

2 Background

This section presents necessary background on collaborative ontology engineering, community-driven ontology matching and an overview of the M-Gov framework.

2.1 Collaborative ontology engineering

Ontology engineering refers to the study of the activities related to the ontology development, the ontology life cycle, and tools and technologies for building the ontologies [6]. In the situation of a collaborative ontology engineering, platforms and tools are designed to help stakeholders to reach a consensus in an asynchronous manner. To facilitate and practice consensus-building in a collaborative environment, the community needs to control each activity, and be able to trace the process and results achieved so far.

In collaborative ontology-engineering, publishing the new version of an ontology is different to a centralized situation, as there is a need to synchronize the editing. To facilitate the editing, web-based or desktop based applications are used, and versions of ontologies are traced with the help of distributed versioning software [6].

In contrast, our approach does not use distributed versioning software for traceability during alignment creation. M-Gov itself keeps track of each activity that occurs in an alignment creation process.

2.2 Community-driven ontology matching

Community-driven ontology matching (CDOM) extends conventional ontology matching by involving the community (end users, knowledge engineers, and developers) in the creation, description, and reuse of mappings [5]. The CDOM is described as a manual task which is based on the following types of information: a) Users: the information about the contributors in the matching process; b) Communities: the information about the relationship among the agents; c) Tools: these tools match the two different ontologies automatically.

A prototype has been implemented and analyzed in [5], which supports the community driven approach. It annotates the community-related information in the basic ontology alignment format. The service has been available online since November 2004. The results show that the acquisition of shared ontology mappings among the web communities is feasible. However, the approach does not annotate the other useful information about the mappings such as "why this mapping seems to be legitimate", etc. This information can serve as the rationale behind a particular mapping.

In contrast, M-Gov captures each activity that occurs during alignment creation. The captured information could serve as the rationale for the creation of a mapping. It also allows one to facilitate the discovery and reuse of existing alignments with the help of queries and thus making the alignment creation process more traceable.

2.3 M-Gov Framework

Governance refers to [9] "*what decisions must be made to ensure effective management and use of IT and who makes the decisions.*" Data governance is required to improve the data quality, which in result improves the maintenance of data [7]. For addressing the data quality issues, [8] suggested to use a holistic approach, which focuses on the people, process, and technology.

[4] uses an extension of PROV-O (metadata) to describe the ontology mapping process, which captures the information of people (stakeholders), process (activities/ discussions), and technology (matcher) as suggested in [8]

A project-centric perspective has been adopted by [4] to deal with the ontology mapping process. The M-Gov framework is based on the project-centric perspective. In the framework, a single ontology mapping project (process) is divided into six phases as follows: **1) Stage:** This phase constitutes the identification of the stakeholders, setting up the scope of the project and enumerate the requirements. **2) Characterize:** It identifies and analyzes the ontologies for generating mappings between them.

As in [10], it is referred as "*to analyze the addressed ontologies to identify difficulties that may be involved for generating mappings.*" **3) Reuse:** It discovers whether any existing alignment can be used for the new mappings. **4) Match:** This phase uses the information captured in the characterization phase. The selected ontologies and the configured matchers are used to identify the potential correspondences, which need to be evaluated for their fitness to form an alignment. **5) Align and Map:** Manual refinement of the candidate correspondences is needed to create an alignment. The rules written based on the alignment is called as mapping. **6) Application:** The stakeholders identify the application, which will use the formed mappings. If either source or target ontologies change over time, this will trigger the new interaction in the community and lead to a new version of mapping.

Adopting a project-centric perspective in ontology mapping process allows one to capture the metadata of various aspects of the mapping process. Using the extension of PROV-O as metadata model makes the ontology mapping process more traceable, as it will not only allow one to formulate queries to reuse existing mappings but also formulate questions about the activities happened during the mapping process.

This paper is built on [4] by a) using an extension of PROV-O to capture each activity in alignment creation process; b) using IBIS [12] for structuring the discussions; c) extending the work done by [4] on M-Gov framework. The "stage" and "characterize" phase of M-Gov was already implemented by [4].

This paper extends the initial M-Gov implementation; it implements the "match phase" of M-Gov and evaluates the correspondences identified in match phase. The next section presents the methodology adopted for ontology matching and evaluation of correspondences.

3 Match Phase of M-Gov framework

This section describes the requirements, design, and implementation of the match phase newly developed for the M-Gov framework.

3.1 Functional requirements

The main objective of the Match Phase is to identify the potential correspondences between two datasets automatically and capture the metadata produced during the alignment creation [4], with the following functional requirements being derived. The match phase should allow a user to configure the matcher by selecting a source ontology, a target ontology, and a matching tool. A matching tool needs to be used to identify the correspondences between the selected ontologies automatically. Identified correspondences need to be displayed on a web page. Users should be allowed to discuss every displayed correspondence with other users by presenting their opinion about its fitness. Based on the discussion, users should be allowed to accept or reject a correspondence. The configuration of matcher, identified correspondence, and discussions of the users about the fitness of the correspondences, need to be stored as the metadata. The metadata should be captured in a queryable format, as that will enable the traceability in the alignment creation process.

3.2 Design

To fulfill the functional requirements, there needed to be a number of aspects designed. In this section, we present a quick overview of the design. The design was focused on an initial baseline without sophisticated UI as our focus was on interaction process and capturing of discussions. Future work will develop the UI. In addition, we focused on an alignment problem where pre-processing is not necessary, as the experimental focus was on traceability of the captured discussions. However, it would be easy to add further steps and linked discussions in the M-Gov framework.

A web based form was built to allow the users to configure the matcher by selecting a source and target ontology, and a matcher tool. The matcher configuration was stored in a database. Selected ontologies were matched using Alignment API 4.8. A REST call was designed for communicating with the Alignment API. The Alignment API returns the potential correspondences in alignment format (an XML format as shown in Fig. 2.), which was used to capture the M-Gov metadata about the identified correspondences. The captured metadata is again stored in the database. Furthermore, an interface was designed to present the M-Gov metadata about the potential correspondences for stakeholders to discuss. To provide context for discussions about the correspondences, the values of object1 and object2 on the interface were linked to their online Linked Data resources. The interface was also designed to show the comments of all the stakeholders on a correspondence. Thus, allows the stakeholders to see other perspectives about the fitness of a correspondence. The discussions of stakeholders are structured by using the IBIS framework and the metadata model used in the M-Gov framework is an extension of PROV-O, as suggested by [4]. Fig. 1 shows the interaction between the elements of the design during the match phase of the M-Gov framework.

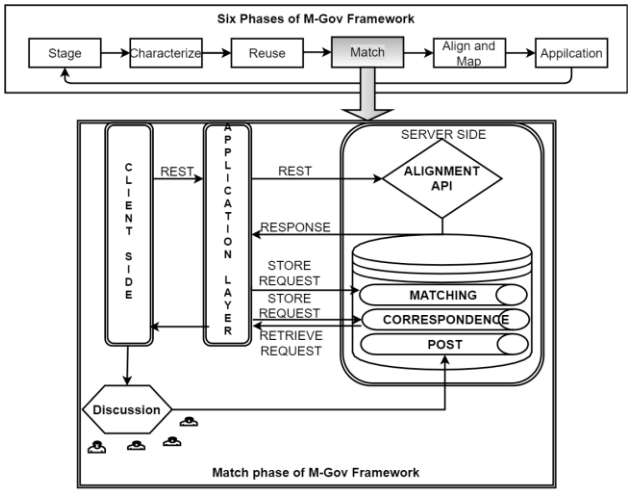


Fig. 1. Design of match phase of M-Gov Framework [4]

The capture of discussions was the major challenge faced while designing the M-Gov match phase supports, as this will enable the traceability in an alignment creation

process. For this, we capture every statement given by each stakeholder during the alignment creation. In M-Gov every statement is linked with its creator, and correspondence's ID on which the statement has been made. M-Gov also captures the conclusion and the stakeholder's ID who concluded that discussion. Table 2 describes the M-Gov metadata used to track the discussion.

Table 1: M-Gov metadata related to discussion

M-Gov captured metadata	Description
discussionID	Unique identifier attached to each discussion
type	Type of discussion: a conclusion or just an opinion
creator	Stakeholder who made the statement
reply	Content of statement
replyType	Type of statement, e.g.: supporting or objecting
conclusion	Final statement while concluding the correspondence
decided	Timestamp of the conclusion
decidedBy	Stakeholder who concluded the correspondence
outcome	If the correspondence is accepted to rejected

3.3 Implementation

A form has been built to allow a user to select a source and target ontology, and a matcher tool. A user can select these parameters from a drop-down menu to configure the matcher. The M-Gov uses these parameters to create the URL to invoke a REST call to Alignment API. Fig. 2 describes the response from Alignment API, it shows an example of a potential equivalence correspondence (line 5) between "HumanActor" (line 3) and "HumanActorAge" (line 4) with a confidence of 0.93 (line 6).

```
<map>
    <cell>
        <entity1 rdf:resource=https://.../thinkhome/ActorOntology.owl#HumanActor/>
        <entity1 rdf:resource=https://.../thinkhome/ActorOntology.owl#HumanActorAge/>
        <relation>=<relation>
        <measure rdf:datatype=http://www.w3.org/2001/XMLSchema#float>0.9347826086956521</measure>
    </cell>
</map>
```

Fig. 2. Response of Alignment API

The M-Gov displays every potential correspondence on a webpage using grid tables, which also contains a "state" column, whose default value is "inDiscussion". The M-Gov also attaches a "change decision" button to every displayed correspondence, which is used to start a new discussion thread for that correspondence. If the discussion thread is already created then this button will lead to the in progress discussion for that correspondence. Once the users reach a consensus after discussion, the M-Gov provides a "Conclude discussion" link, which allows a user to change the state of the correspondence to either "Accepted" or "Rejected". The M-Gov also stores the discussions along with the user's information under the "post" table in the database.

Fig. 3 represents the page by which stakeholders can add their arguments to participate in a discussion about a correspondence. In our example of Fig. 2, this would involve discussion of whether HumanActor and HumanActorAge are really equivalent? Fig. 3 shows the overview of the correspondence and arguments about its fitness. "reply" textbox can be used to add arguments, while a suitable reply type needs to be selected from the dropdown "Reply Type", whose values are "Supporting example, objecting example, supporting justification, objecting justification, supporting motivation and objecting motivation".

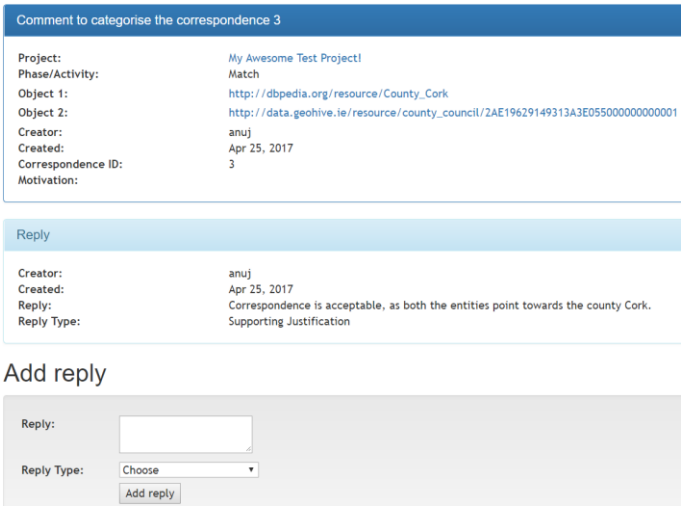


Fig. 3. M-Gov Match Discussion page

4 Evaluation

Motivation. The purpose of this experiment was to trace the discussions among the stakeholders during the alignment creation process and identify the following: 1) level of participation of various users of a group during alignment creation. 2) most discussed correspondences by users of a group. 3) accuracy of a group in creating alignment.

In the experiment, we have used three types of correspondences: a) Correct correspondences - those in which both objects point towards the same resource. b) Incorrect correspondences - those in which both objects point towards completely different resources. c) Ambiguous correspondences - those in which both objects point towards different resources. But to understand the difference, a user needs to go through a substantial amount of information, as the difference might not be clear from the label of the entities.

Hypothesis. In most cases, the discussion thread attached to an ambiguous correspondence will be longer than correct and incorrect correspondences.

Experiment method. We formed 4 groups, 3 groups contained 3 users while 1 group contained only 2 users. A separate instance of the framework was provided for

each group. Every user was located at a different workstation and was allocated discrete credentials to log into the framework. We have only used instance level correspondences in the experiment, since creating concept level correspondences requires participants with a deeper understanding (who are harder to recruit). It was thus decided to first investigate stakeholder collaboration tracing using instance level correspondences, which could be performed by a wider range of participants. We created a discrete set of 7 instance level equivalence correspondences for each group, the complete list is available online[1]. Semantic mapping researchers validated the created correspondences. These correspondences have been created manually and injected in the framework for discussion, which covers three types of correspondences as follows: a) Correct correspondence: These are created by taking an entity from OSi[2] dataset as object1, while the object2 has been selected from DBpedia[3], which points to the exact same resource as referred by object1. For example, "County Roscommon represented by OSi" and "County Roscommon represented by DBpedia", b) Incorrect correspondence: These are created by taking an entity from OSi dataset as object1, while the object2 has been selected from DBpedia, which points to a completely different resource than that referred by object1. For example, "County Roscommon represented by OSi" and "County Clare represented by DBpedia", c) Ambiguous correspondence: These are created by taking an entity from OSi dataset as object1, while the object2 has been selected from DBpedia, which points to the resource that has a label similar to the resource referred by object1. To figure out the difference between both objects, a user needs to examine the available information about both the resources. For example, "County Tipperary represented by OSi" and "Tipperary town represented by DBpedia". Participants can discuss the correspondences within the group only through the framework. For deciding upon a correspondence, if it was acceptable or not, users needed to come to a consensus.

Metrics. To trace the most discussed correspondences in a group, the word count in the statements of the users will be used to calculate the length of the discussion. The word count for a discussion in a group also depends on the active users in a group. At the end of the experiment, users will be asked to evaluate the use of the framework by providing answers to usability based questions of PSSUQ [11].

Datasets. A subset of entities in the OSi county dataset and in the DBpedia dataset for counties of the Republic of Ireland has been used to create correspondences.

User recruitment. The selected users were M.Sc. students of computer science at Trinity College Dublin. For preparing the users for the experiment, we have given a presentation, a video tutorial and a detailed version of user instructions to users about how to use the M-Gov to curate the correspondences. All the documents related to the experiment preparation are available online[4].

Data analysis. For each group, Fig. 4 describes the type of correspondence and length of discussion involved in coming to the conclusion. Fig. 5 describes the individual contribution of the users in each group. Group 1 had 3 users: user 9, 10, and 11. However, user 10 did not participate in the discussion properly. For group 1, the

[1] https://github.com/anujsinghdm/Experiment/blob/master/AllCorrepondence.xlsx
[2] http://data.geohive.ie/
[3] http://wiki.dbpedia.org/
[4] https://github.com/anujsinghdm/Experiment/tree/master/UserInstruction

longest discussion thread has been attached to C4, which is a correct correspondence and it is also clear from Fig. 4 and 5, user 9 and 11 were mostly discussing the non-ambiguous correspondences, hence group 1 does not support the hypothesis.

Group 2 had 2 users: user 7 and 8. However, the majority of the word count represents the user 8. For group 2, the longest discussion threads have been attached to C1 and C4, where C1 is the correct correspondence, while the C4 is the ambiguous correspondence. Users did not discuss much the 2nd ambiguous correspondence, only user 8 gave the statement, why it wants to reject the correspondence. Having the discussions analyzed, we can say that user 7 did not participate in the discussions properly and group 2 is also not in support of the hypothesis.

Group 3 had 3 users: user 4, 5, and 6. For group 3, the longest discussion thread has been attached to an ambiguous correspondence C4. Fig. 4 describes that group 3 discussed incorrect correspondences more. This might be the reason why the 2nd ambiguous correspondence does not have a longer discussion, as the users perceived it a correct correspondence. Group 3 concluded one more correspondence incorrectly, but we believe that is just an operation error, as the attached discussion indicates that they analyzed the correspondence correctly. Group 3 supports the hypothesis as the longest discussion thread is attached to an ambiguous correspondence.

Group 4 had 3 users: user 1, 2, and 3. However, most of the discussions have been carried out by user 1. As it is clear from Fig. 4, ambiguous correspondences (C4 and C7) have the longest discussion thread attached to them. Hence, group 4 supports the hypothesis.

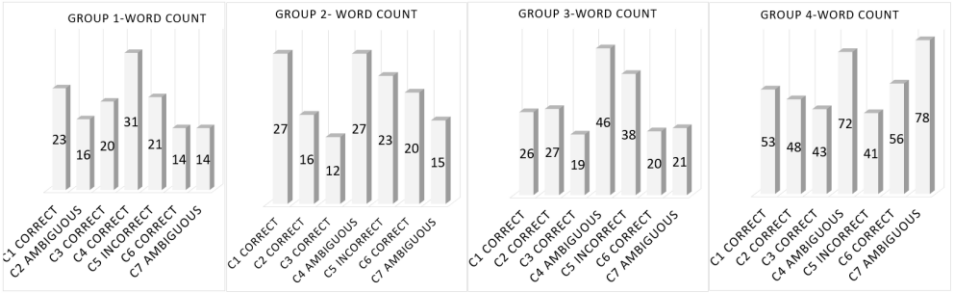


Fig. 4. Word count for correspondences discussed by each group

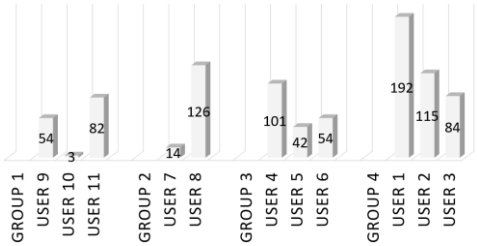


Fig. 5. Individual contribution of each user in discussing the correspondences

Finally, participants were asked to complete a PSSUQ [11] questionnaire. The information in Fig. 6 has been produced by taking the means and standard deviation of the responses of each participant per questions. Then we checked below:

resultant = abs (value (response of a specific user for selected question) - means (responses of all users for the selected question))

if the resultant is greater than the standard deviation (responses of all users for a specific question), we marked false for that specific response. Finally, we counted all the "True" values of a specific user for all the questions.

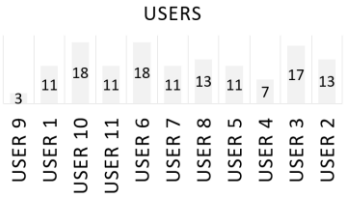


Fig. 6. True responses per user

Conclusions. The results indicate that except group 3, every other group was accurate (compared to the gold standard) in creating alignment. Group 3 incorrectly concluded 2 correspondences out of 7. The results also show that group 1 and 2 do not support the hypothesis, however not every member of this group actively took part in the discussion. Group 3 supports the hypothesis as for the 1st ambiguous correspondence, the discussion thread is the longest. We believe that the users of this group miscomprehended the information of 2nd ambiguous correspondence, hence the correspondence did not get discussed in detail and concluded incorrectly. Group 4 clearly supports the hypothesis as they discussed ambiguous correspondences the most. Gathered data is unable to lead us to any conclusion about the hypothesis, as two groups are supporting the hypothesis while the other two groups are not in support of it. However, the results provide an evidence that the captured metadata by M-Gov enabled the traceability in the alignment creation process. Additionaly, for the technical contribution we tracked the following: a) level of participation of users, b) most discussed correspondences and, c) the accuracy of groups in alignment creation. We can also conclude from Fig. 6, user 9 and 4 are the outliers as most of their responses do not comply with other users. The data from the PSSUQ suggests that 72% users were satisfied by using M-Gov but enhancements in terms of UI/UX are required so that tasks could be performed more efficiently.

5 Related work

A variety of approaches has been used to evaluate the methodologies/ frameworks that support collaborative ontology engineering. We see two evaluation approaches related to our work. This section focuses on these approaches.

Domain experts are supported by [13] to engineer an ontology in a distributed environment. In the start of the process, an initial version of an ontology needs to be

created by users then they can use it and locally adapt it for their own purpose. There is no support to change the ontology shared by all the users, only control board handles the changes to a shared ontology. The board deploys the feasible changes in the next version. [13] also describes a two stage experiment for a creating an ontology. In the first stage, users argued for a change without any guidelines, while in second stage they were given a subset of the arguments that had been found effective in stage one of the experiment. The paper concluded that the creation of ontology proceeded faster during the second stage. We could benefit from [13] in our future work by giving some more restricted guidelines to the users for a discussion.

The Ontology development framework proposed in [14] supports various users to reach consensus through iterative evaluations. [15] describes a consensus based experiment using [14]. 7 users were involved in the experiment, which are of different competency. The coordinator has created an initial version of an ontology. Iterative evaluation is done by each user by an "initial ontology evaluation sheet" that helps to evolve the ontology. They used Nominal Group Technique (NGT) for the evaluation. In contrast, we support online discussion among users located at different locations. Our approach also captures the discussions to enable the traceability in the alignment creation process. [15] uses the degree of participation (dop), which is leveraged by the facilitator to determine the quality of an ontology. In contrast, we have measured the dop by word count in the statements of each user during the discussion. We have noticed in our experiment that the groups in which the users were more active are supporting the hypothesis formed for the experiment.

6 Conclusion

The paper presents an extension of M-Gov framework to match the two different datasets automatically and capture the metadata produced during the alignment creation. The paper also describes an experiment in which 11 stakeholders discussed the potential correspondences to create an alignment. The aim was to trace the metadata produced during the alignment creation.

The research question presented in this paper is to what extent the captured metadata allows us to trace the most discussed correspondences by users, the level of participation of users, and the decisions undertaken by a group of users for a correspondence to determine if it is acceptable or not, in an alignment creation process? We also present the evaluation of M-Gov by users in creating alignment.

An experiment was conducted to create an alignment between the locations in DBpedia and OSi dataset. Based on the results, we are unable to conclude the hypothesis, as two groups are supporting it while the other two groups are not in support of the hypothesis. However, it provides an evidence that the captured metadata during the alignment creation enables traceability. In addition to this, the technical contribution of our work involves tracing the following: a) Group 1 and 2 discussed mostly the non-ambiguous correspondences, as the discussion thread attached to non-ambiguous correspondences are the longest. Group 3 and 4 have the longest thread attached to the ambiguous correspondences, so group 3 and 4 discussed mostly the ambiguous correspondences. b) Not every participant in group 1 and 2 was actively

engaged in the discussion. c) 26 correspondences out of 28 were concluded correctly. We would be able to do more detailed analysis if the participants would have been more active in each group as we would have got richer experiment data.

Acknowledgements. This research has received funding from the ADAPT Centre for Digital Content Technology, funded under the SFI Research Centres Programme (Grant 13/RC/2106) and co-funded by the European Regional Development Fund.

References

1. Silva, N. and Rocha, J., 2003. Service-oriented ontology mapping system. In Semantic Integration Workshop (SI-2003) (p. 144).
2. Choi, N., Song, I.Y. and Han, H., 2006. A survey on ontology mapping. ACM Sigmod Record, 35(3), pp.34-41.
3. Gotel, O.C. and Finkelstein, C.W., 1994. An analysis of the requirements traceability problem. Proceedings of IEEE International Conference on Requirements Engineering (pp. 94-101). IEEE.
4. Debruyne, C., Walshe, B. and O'Sullivan, D., 2015. Towards a project centric metadata model and lifecycle for ontology mapping governance. In Proceedings of the 17th International Conference on Information Integration and Web-based Applications & Services (p. 50). ACM.
5. Zhdanova, A. and Shvaiko, P., 2006. Community-driven ontology matching. The Semantic Web: Research and Applications, pp.34-49.
6. Simperl, E. and Luczak-Rösch, M., 2014. Collaborative ontology engineering: a survey. The Knowledge Engineering Review, 29(1), pp.101-131.
7. Cheong, L.K. and Chang, V., 2007. The need for data governance: a case study. ACIS 2007 Proceedings, p.100.
8. Friedman, T., 2006. Key issues for data management and integration, 2006. Gartner Research.
9. Khatri, V., and Carol V.B., 2010. Designing data governance. Communications of the ACM, 53(1), pp.148-152.
10. Thomas, H., Brennan, R. and O'Sullivan, D., 2011. MooM--a prototype framework for management of ontology mappings. IEEE International Conference on Advanced Information Networking and Applications (pp. 548-555). IEEE.
11. Fruhling, A. and Lee, S., 2005. Assessing the reliability, validity and adaptability of PSSUQ. AMCIS 2005 Proceedings, p.378.
12. Kunz, W. and Rittel, H.W., 1972. Information science: on the structure of its problems. Information Storage and Retrieval, 8(2), pp.95-98.
13. Pinto, H.S., Staab, S. and Tempich, C., 2004, August. DILIGENT: Towards a fine-grained methodology for DIstributed, Loosely-controlled and evolvInG Engineering of oNTologies. In Proceedings of the 16th European Conference on Artificial Intelligence (pp. 393-397). IOS Press.
14. Holsapple, C.W. and Joshi, K.D., 2002. A collaborative approach to ontology design. Communications of the ACM, 45(2), pp.42-47.
15. Karapiperis, S. and Apostolou, D., 2006. Consensus building in collaborative ontology engineering processes. Journal of Universal Knowledge Management, 1(3), pp.199-216.
16. Ehrig, M. and Staab, S., 2004. QOM-quick ontology mapping. In Third International Semantic Web Conference (pp. 683-697). Springer.

Results of the Ontology Alignment Evaluation Initiative 2017*

Manel Achichi[1], Michelle Cheatham[2], Zlatan Dragisic[3], Jérôme Euzenat[4], Daniel Faria[5], Alfio Ferrara[6], Giorgos Flouris[7], Irini Fundulaki[7], Ian Harrow[8], Valentina Ivanova[3], Ernesto Jiménez-Ruiz[9], Kristian Kolthoff[10], Elena Kuss[10], Patrick Lambrix[3], Henrik Leopold[11], Huanyu Li[3], Christian Meilicke[10], Majid Mohammadi[12], Stefano Montanelli[6], Catia Pesquita[13], Tzanina Saveta[7], Pavel Shvaiko[14], Andrea Splendiani[8], Heiner Stuckenschmidt[10], Elodie Thiéblin[15], Konstantin Todorov[1], Cássia Trojahn[15], and Ondřej Zamazal[16]

[1] LIRMM/University of Montpellier, France
lastname@lirmm.fr
[2] Data Semantics (DaSe) Laboratory, Wright State University, USA
michelle.cheatham@wright.edu
[3] Linköping University & Swedish e-Science Research Center, Linköping, Sweden
{zlatan.dragisic,valentina.ivanova,patrick.lambrix,huanyu.li}@liu.se
[4] INRIA & Univ. Grenoble Alpes, Grenoble, France
Jerome.Euzenat@inria.fr
[5] Instituto Gulbenkian de Ciência, Lisbon, Portugal
dfaria@igc.gulbenkian.pt
[6] Università degli studi di Milano, Italy
{alfio.ferrara,stefano.montanelli}@unimi.it
[7] Institute of Computer Science-FORTH, Heraklion, Greece
{jsaveta,fgeo,fundul}@ics.forth.gr
[8] Pistoia Alliance Inc., USA
{ian.harrow,andrea.splendiani}@pistoiaalliance.org
[9] Department of Informatics, University of Oslo, Norway
ernestoj@ifi.uio.no
[10] University of Mannheim, Germany
{christian,elena,heiner}@informatik.uni-mannheim.de
[11] Vrije Universiteit Amsterdam, Netherlands
h.leopold@vu.nl
[12] Faculty of Technology, Policy, and Management, Technical University of Delft, Netherlands
m.mohammadi@tudelft.nl
[13] LASIGE, Faculdade de Ciências, Universidade de Lisboa, Portugal
cpesquita@di.fc.ul.pt
[14] TasLab, Informatica Trentina, Trento, Italy
pavel.shvaiko@infotn.it
[15] IRIT & Université Toulouse II, Toulouse, France
{cassia.trojahn}@irit.fr
[16] University of Economics, Prague, Czech Republic
ondrej.zamazal@vse.cz

Abstract. Ontology matching consists of finding correspondences between semantically related entities of different ontologies.

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

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 2017 campaign offered 9 tracks with 23 test cases, and was attended by 21 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 [20, 22]. 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, tool developers can improve their systems.

Two first events were organized in 2004: *(i)* the Information Interpretation and Integration Conference (I3CON) held at the NIST Performance Metrics for Intelligent Systems (PerMIS) workshop and *(ii)* the Ontology Alignment Contest held at the Evaluation of Ontology-based Tools (EON) workshop of the annual International Semantic Web Conference (ISWC) [46]. 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) [5]. From 2006 until the present, the OAEI campaigns were held at the Ontology Matching workshop, collocated with ISWC [2, 3, 7–9, 13, 16–19, 21], which this year took place in Vienna, Austria[2].

Since 2011, we have been using an environment for automatically processing evaluations (§2.2) 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. In the OAEI 2017, a novel evaluation environment called HOBBIT (§10) was adopted for the novel HOBBIT Link Discovery track. Except for this track, all systems were executed under the SEALS client in all other tracks. The Benchmark track was discontinued in this edition of the OAEI.

This paper synthesizes the 2017 evaluation campaign and introduces the results provided in the papers of the participants. The remainder of the paper is organised as follows: in Section 2, we present the overall evaluation methodology that has been used; Sections 3-11 discuss the settings and the results of each of the test cases; Section 13 overviews lessons learned from the campaign; and finally, Section 14 concludes the paper.

[1] http://oaei.ontologymatching.org
[2] http://om2017.ontologymatching.org
[3] http://www.seals-project.eu

2 General methodology

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

2.1 Tracks and test cases

This year's OAEI campaign consisted of 9 tracks gathering 23 test cases, and different evaluation modalities:

Expressive Ontology tracks offer alignments between real world ontologies expressed in OWL:

- **Anatomy (§3):** The anatomy track comprises a single test case consisting of matching the Adult Mouse Anatomy (2744 classes) and a small fragment of the NCI Thesaurus (3304 classes) describing the human anatomy. Results are evaluated automatically against a manually curated reference alignment.
- **Conference (§4):** The conference track comprises a single test case that is a suite of 21 matching tasks corresponding to the pairwise combination of 7 ontologies describing the domain of organizing conferences. Results are evaluated automatically against reference alignments in several modalities, and by using logical reasoning techniques.
- **Large biomedical ontologies (§5):** The largebio track comprises 6 test cases involving 3 large and semantically rich biomedical ontologies: FMA, SNOMED-CT, and NCI Thesaurus. These test cases correspond to the pairwise combination of these ontologies in two variants: small overlapping fragments, in which only overlapping sections of the ontologies are matched, and whole ontologies. The evaluation is based on reference alignments automatically derived from the UMLS Metathesaurus, with mappings causing logical incoherence flagged so as not to be taken into account.
- **Disease & Phenotype (§6):** The disease & phenotype track comprises 4 test cases that involve 6 biomedical ontologies covering the disease and phenotype domains: HPO versus MP, DOID versus ORDO, HPO versus MeSH, and HPO versus OMIM. The evaluation has been performed according to (1) a consensus alignment generated from those produced by the participating systems, (2) a set of manually generated mappings, and (3) a manual assessment of unique mappings (i.e., mappings that are not suggested by other systems).

Multilingual tracks offer alignments between ontologies in different languages:

- **Multifarm (§7):** The multifarm track is based on a subset of the Conference data set translated into ten different languages, in addition to their original English: Arabic, Chinese, Czech, Dutch, French, German, Italian, Portuguese, Russian, and Spanish. It consists of two test cases: same ontologies, where two versions of the same ontology in different languages are matched, and different ontologies, in which two different ontologies in different languages are matched. In

total, 45 language pairings are evaluated, meaning that the same ontologies test case comprises 315 matching tasks, and the different ontologies test case comprises 945 matching tasks. Results are evaluated automatically against reference alignments.

Interactive tracks provide simulated user interaction to enable the benchmarking of algorithms designed to make use of it, with respect to both the improvement in the results and the workload of the user:

Interactive Matching Evaluation (§8): The Interactive track is based on the test cases from the anatomy and conference tracks. An Oracle, which matching tools can access programmatically, simulates user feedback by querying the reference alignment of the test case. The Oracle can generate erroneous responses at a given rate, to simulate user errors. The evaluation is based on the same reference alignments, and contemplates the number of user interactions and the fraction of erroneous responses received by the tool, in addition to the standard evaluation parameters.

Instance Matching tracks focus on alignments between ontology instances expressed in the form of OWL Aboxes:

Instance Matching (§9). The instance track comprises two independent sub-tracks:

SYNTHETIC: This sub-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). It includes two evaluation modalities, *Sandbox* and *Mainbox*, which differ on the number of instances to match. The evaluation is automatic, based on a reference alignment, and partially blind – matching tools have access only to the *Sandbox* reference alignment.

DOREMUS: This sub-track consists of matching real world datasets about classical music artworks from two major French cultural institutions: the French National Library (BnF) and the Philharmonie de Paris (PP). Both datasets use the same vocabulary, the DOREMUS model, issued from the DOREMUS project[4]. This sub-track comprises two different test cases called *heterogeneities* (HT) and *false-positives trap* (FPT) characterized by different degrees of heterogeneity in artwork descriptions. The evaluation is automatic and based on reference alignments.

HOBBIT Link Discovery (§10). The HOBBIT track aims to deal with link discovery for spatial data represented as trajectories or traces i.e., sequences of longitude, latitude pairs. It comprises two test cases: Linking and Spatial. The Linking test case consists in matching traces that have been modified using string-based approaches, different date and coordinate formats, and by addition and/or deletion of intermediate points. In the Spatial test case, the goal is to identify DE-9IM (Dimensionally Extended nine-Intersection Model) topological relations between traces: *Equals*, *Disjoint*, *Touches*, *Contains/Within*, *Covers/CoveredBy*, *Intersects*, *Crosses*, *Overlaps*. For each relation, a different pair of source and target datasets is given to the participants, so the test

[4] `http://www.doremus.org`

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

test	formalism	relations	confidence	modalities	language	SEALS
anatomy	OWL	=	[0 1]	open	EN	√
conference	OWL	=, <=	[0 1]	open+blind	EN	√
largebio	OWL	=	[0 1]	open	EN	√
phenotype	OWL	=	[0 1]	blind	EN	√
multifarm	OWL	=	[0 1]	open+blind	AR, CZ, CN, DE, EN, ES, FR, IT, NL, RU, PT	√
interactive	OWL	=, <=	[0 1]	open	EN	√
instance	OWL	=	[0 1]	open+blind	EN	√
HOBBIT	OWL	=, spatial	N/A	open+blind	EN, N/A	
process model	OWL	<=	[0 1]	open+blind	EN	√

case consists of 8 individual matching tasks. In both test cases, two evaluation modalities, *Sandbox* and *Mainbox*, were considered, differing on the number of instances to match. The evaluation is automatic and based on reference alignments.

Process Model Matching (§11): The process model track is concerned with the application of ontology matching techniques to the problem of matching process models. It comprises two test cases used in the Process Model Matching Campaign 2015 [4] which have been converted to an ontological representation, with process model entities being represented as ontology instances. The first test case contains nine process models which represent the application process for a master program of German universities as well as reference alignments between all pairs of models. The second test case consists of process models which describe the process of registering a newborn child in different countries. The evaluation is automatic, based on reference alignments, and uses standard precision and recall measures as well as a probabilistic variant described in [29].

Table 1 summarizes the variation in the proposed test cases.

2.2 The SEALS client

Since 2011, tool developers had to implement a simple interface and to wrap their tools in a predefined way including all required libraries and resources. A tutorial for tool wrapping was provided to the participants, describing how to wrap a tool and how to use the SEALS client to run a full evaluation locally. This client is then executed by the track organizers to run the evaluation. This approach ensures the reproducibility and comparability of the results of all systems.

2.3 Preparatory phase

Ontologies to be matched and (where applicable) reference alignments have been provided in advance during the period between June 1^{st} and July 15^{th}, 2017. This gave potential participants the occasion to send observations, bug corrections, remarks and other test cases to the organizers. The goal of this preparatory period is to ensure that the delivered tests make sense to the participants. The final test base was released on July 15^{th}, 2017 and did not evolve after that.

2.4 Execution phase

During the execution phase, participants used their systems to automatically match the test case ontologies. In most cases, ontologies are described in OWL-DL and serialized in the RDF/XML format [11]. Participants can self-evaluate their results either by comparing their output with reference alignments or by using the SEALS client to compute precision and recall. They can tune their systems with respect to the non blind evaluation as long as the rules published on the OAEI web site are satisfied. This phase has been conducted between July 15^{th} and August 31^{st}, 2017, except for the HOBBIT track which was extended until September 15^{th}, 2017. Like last year, we requested a mandatory registration of systems and a preliminary evaluation of wrapped systems by July 31st, to alleviate the burden of debugging systems with respect to issues with the SEALS client during the Evaluation phase.

2.5 Evaluation phase

Participants were required to submit their SEALS-wrapped tools by August 31^{st}, 2017, and their HOBBIT-wrapped tool by September 15^{th}, 2017. Tools were then tested by the organizers and minor problems were reported to some tool developers, who were given the opportunity to fix their tools and resubmit them.

Initial results were provided directly to the participants between September 1^{st} and October 15^{th}, 2017. The final results for most tracks were published on the respective pages of the OAEI website by October 15^{th}, although some tracks were delayed.

The standard evaluation measures are precision, recall and F-measure computed against the reference alignments. More details on the evaluation are given in the sections for the test cases.

2.6 Comments on the execution

Following an initial period of growth, the number of OAEI participants has remained approximately constant since 2012, at slightly over 20 (see Figure 1). This year was no exception, as we counted 21 participating systems. Table 2 lists the participants and the tracks in which they competed. Some matching systems participated with different variants (DiSMatch and LogMap) whereas others were evaluated with different configurations, as requested by developers (see test case sections for details).

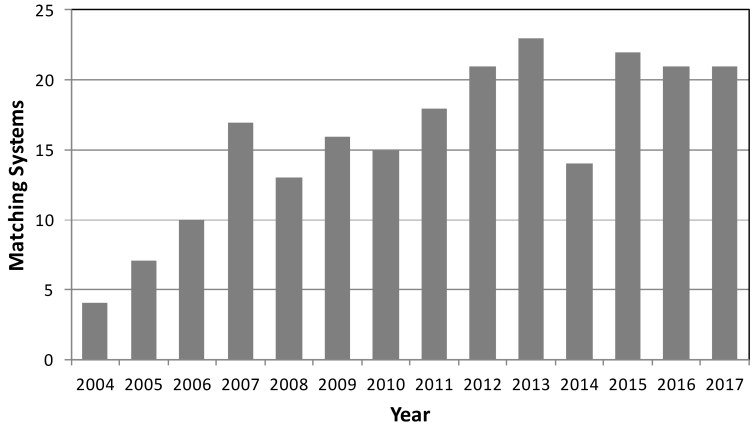


Fig. 1. Number of participating systems per year in the OAEI.

Table 2. Participants and the status of their submissions.

System	ALIN	AML	CroLOM	DiSMatch-ar	DiSMatch-sg	DiSMatch-tr	I-Match	KEPLER	Legato	LogMap	LogMap-Bio	LogMapLt	njuLink	ONTMAT	POMap	RADON	SANOM	Silk	Wiki2	XMap	YAM-BIO	Total=21
Confidence	✓	✓	✓	✓	✓	✓	-	✓	✓	✓	✓	-	✓	-	✓	✓	✓	✓	-	✓	-	16
anatomy	●	●	○	○	○	○	○	●	○	●	●	●	○	○	●	○	●	○	●	●	●	11
conference	●	●	○	○	○	○	○	●	○	●	○	●	○	●	●	○	●	○	●	●	○	10
largebio	○	●	○	○	○	○	○	◐	○	●	●	●	○	○	◐	○	◐	○	◐	●	●	10
phenotype	○	●	○	●	●	●	○	◐	○	●	●	◐	○	○	◐	○	○	○	○	◐	◐	11
multifarm	○	●	●	○	○	○	○	●	○	●	○	○	○	○	○	○	◐	○	●	◐	○	7
interactive	●	●	○	○	○	○	○	○	○	●	○	○	○	○	○	○	○	○	○	●	○	4
process model	○	●	○	○	○	○	●	○	○	●	○	○	○	○	○	○	○	○	○	○	○	3
instance	○	●	○	○	○	○	●	○	●	●	○	○	◐	○	○	○	○	○	○	○	○	5
hobbit ld	○	●	○	○	○	○	●	○	○	○	○	○	○	○	○	◐	○	◐	○	○	○	4
total	3	9	1	1	1	1	3	5	1	8	3	4	1	1	4	1	4	1	4	6	3	65

Confidence pertains to the confidance 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.

3 Anatomy

The anatomy test case confronts matching systems with two fragments of biomedical ontologies which describe the human anatomy[5] and the anatomy of the mouse[6]. This data set has been used since 2007 with some improvements over the years [15].

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

3.1 Experimental Setting

We conducted experiments by executing each system in its standard setting and we compare precision, recall, F-measure and recall+ against a manually curated reference alignment. Recall+ indicates the amount of detected non-trivial correspondences, i.e., correspondence that do not have the same normalized label. The approach that generates only trivial correspondences is depicted as baseline StringEquiv in the following section.

We ran the systems on a server with 3.46 GHz (6 cores) and 8GB allocated RAM, using the SEALS client. However, we changed the way precision and recall are computed by removing trivial correspondences in the oboInOwl namespace like:

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

as well as correspondences expressing relations different from equivalence. Thus, the results generated by the SEALS client vary in some cases by 0.5% compared to the results presented below. Using the Pellet reasoner we also checked whether the generated alignment is coherent, i.e., that there are no unsatisfiable classes when the ontologies are merged with the alignment.

3.2 Results

In Table 3, we show the results of the 11 participating systems that generated an alignment, including 3 versions of LogMap. A number of systems participated in the anatomy track for the first time this year: KEPLER, POMap, SANOM, WikiV2, and YAM-BIO. For more details, we refer the reader to the papers presenting the systems.

Table 3. Comparison, ordered by F-measure, against the reference alignment, runtime is measured in seconds, the "size" column refers to the number of correspondences in the generated alignment.

Matcher	Runtime	Size	Precision	F-measure	Recall	Recall+	Coherent
AML	47	1493	0.95	0.943	0.936	0.832	√
YAM-BIO	70	1474	0.948	0.935	0.922	0.794	-
POMap	808	1492	0.94	0.933	0.925	0.824	-
LogMapBio	820	1534	0.889	0.894	0.899	0.733	√
XMap	37	1412	0.926	0.893	0.863	0.639	√
LogMap	22	1397	0.918	0.88	0.846	0.593	√
KEPLER	234	1173	0.958	0.836	0.741	0.316	-
LogMapLite	19	1148	0.962	0.829	0.728	0.29	-
SANOM	295	1304	0.895	0.828	0.77	0.419	-
Wiki2	2204	1260	0.883	0.802	0.734	0.356	-
StringEquiv	-	946	0.997	0.766	0.622	0.000	-
ALIN	836	516	0.996	0.506	0.339	0.0	√

This year 5 out of 11 systems were able to achieve the alignment task in less than 100 seconds: LogMapLite, LogMap, XMap, AML and YAM-BIO. In 2016 and 2015, there

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

were 4 out of 13 systems and 6 out of 15 systems respectively that generated an alignment in this time frame. As in the last 5 years LogMapLite has the shortest runtime. The table shows that there is no correlation between the quality of the generated alignment in terms of precision and recall and the runtime. This result had also been observed in previous OAEI campaigns.

The table also shows the results for F-measure, recall+ and the size of alignments. Regarding F-measure, the top 3 ranked systems AML, YAM-BIO, POMap achieve on F-measure above 0.93. Among these, AML achieved the highest F-measure (0.943). All of the long-term participants in the track showed comparable results in terms of F-measure to their last year's results and at least as good as the results of the best systems in OAEI 2007-2010. Regarding recall+, AML, LogMap, LogMapLite showed similar results to previous years. LogMapBio has a slight increase from 0.728 in 2016 to 0.733 in 2017. XMap decreases a bit from 0.647 to 0.639. Two new participants obtained good results for recall+, POMap scored 0.824 (second place) followed by YAM-BIO with 0.794 (third place). In terms of the number of correspondences, long-term participants computed similar numbers of correspondences as last year. AML and LogMap generated the same number of correspondences, LogMapBio generated 3 more correspondences, LogMapLite generated 1 more, ALIN generated 6 more and XMap generated 1 less.

This year, 10 out of 11 systems achieved an F-measure higher than the baseline. This is a slightly better result than last year when 9 out of 13 surpassed the baseline. Five systems produced coherent alignments, which is comparable to the last two years when 7 out of 13 and 5 out of 10 systems achieved this. Two of the three best systems with respect to F-measure (YAM-BIO and POMap) produced incoherent alignments.

3.3 Conclusions

The number of systems participating in the anatomy track has varied throughout the years. This year, it is lower than in the two previous editions, but higher than in 2014. As noted previously there are newly-joined systems as well as long-term participants.

The systems that participated in the previous edition in 2016 scored similarly to their previous results. As last year, the AML system set the top result for anatomy track with respect to F-measure. Two of the newly-joined systems (YAM-BIO and POMap) achieved 2nd and 3rd best score in terms of F-measure.

4 Conference

The conference test cases require matching several moderately expressive ontologies from the conference organisation domain.

4.1 Test data

The data set consists of 16 ontologies in the domain of organising conferences. These ontologies were developed within the OntoFarm project[7].

The main features of this test case are:

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

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

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

4.2 Results

We performed three kinds of evaluations. First, we provide results in terms of F-measure, comparison with baseline matchers and results of matchers from previous OAEI editions and precision/recall triangular graph based on sharp reference alignments. Second, we provide an evaluation based on the uncertain version of the reference alignment, and finally we also provide an evaluation based on violations of consistency and conservativity principles.

Evaluation based on sharp reference alignments We evaluated the results of participants against blind reference alignments (labelled as *rar2*).[8] This includes all pairwise combinations between 7 different ontologies, i.e., 21 alignments.

We have prepared the reference alignments in two steps. First, we have generated them as a transitive closure computed on the original reference alignments. In order to obtain a coherent result, conflicting correspondences, i.e., those causing unsatisfiability, have been manually inspected and incoherency has been resolved by evaluators. The resulting reference alignments are labelled as *ra2*. Second, we detected violations of conservativity using the approach from [44] and resolved them by an evaluator. The resulting reference alignments are labelled as *rar2*. As a result, the degree of correctness and completeness of the new reference alignments is probably slightly better than for the old one. However, the differences are relatively limited. Whereas the new reference alignments are not open, the old reference alignments (labeled as *ra1* on the conference web page) are available. These represent close approximations of the new ones.

Table 4 shows the results of all participants with regard to the reference alignment *rar2*. $F_{0.5}$-measure, F_1-measure and F_2-measure are computed for the threshold that provides the optimal F_1-measure. F_1 is the harmonic mean of precision and recall where both are equally weighted; F_2 weights recall higher than precision and $F_{0.5}$ weights precision higher than recall. The matchers shown in the table are ordered according to their highest average F_1-measure. We employed two baseline matchers. edna (string edit distance matcher) was used within the benchmark test cases in previous years and with regard to performance it is very similar as the previously used *baseline2* in the conference track; StringEquiv is used within the anatomy test case. This year these baselines divide matchers into two performance groups.

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

Table 4. 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.

Matcher	Prec.	$F_{0.5}$-m.	F_1-m.	F_2-m.	Rec.	Inc.Align.	Conser.V.	Consist.V.
AML	0.78	0.74	0.69	0.65	0.62	0	39	0
LogMap	0.77	0.72	0.66	0.6	0.57	0	25	0
XMap	0.78	0.72	0.65	0.58	0.55	1	22	4
LogMapLt	0.68	0.62	0.56	0.5	0.47	5	96	25
edna	*0.74*	*0.66*	*0.56*	*0.49*	*0.45*			
KEPLER	0.67	0.61	0.55	0.49	0.46	12	123	159
WikiV3	0.63	0.59	0.54	0.5	0.47	10	125	58
StringEquiv	*0.76*	*0.65*	*0.53*	*0.45*	*0.41*			
POMap	0.69	0.59	0.49	0.42	0.38	0	1	0
ALIN	0.86	0.6	0.41	0.31	0.27	0	0	0
SANOM	0.8	0.56	0.38	0.29	0.25	1	11	18
ONTMAT	0.06	0.07	0.1	0.19	0.41	0	1	0

With regard to the two baselines, we can group tools according to each matcher's position. In all, four tools outperformed both baselines (AML, LogMap, XMap and LogMapLt), and two newcomers (KEPLER and WikiV3) performed better than one baseline. Other matchers (POMap, ALIN, SANOM and ONTMAT) performed worse than both baselines. Four tools (ALIN, POMap, ONTMAT and SANOM) did not match properties at all. Of course, this had a negative effect on those tools' overall performance. More details about evaluation considering only classes or properties are on the conference web page. The performance of all matchers (except ONTMAT) regarding their precision, recall and F_1-measure is visualised in Figure 2. Matchers are represented as squares or triangles. Baselines are represented as circles.

Comparison with previous years with regard to rar2 Four matchers, top-performers, also participated in the Conference test cases in OAEI 2016. None of them improved with regard to F1-measure evaluation.

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

There are two ways that we can evaluate matchers according to these "uncertain" reference alignments, which we refer to as *discrete* and *continuous*. The discrete evaluation considers any match in the reference alignment with a confidence value of 0.5 or greater to be fully correct and those with a confidence less than 0.5 to be fully incorrect. Similarly, a matcher's match is considered a "yes" if the confidence value is greater than or equal to the matcher's threshold and a "no" otherwise. In essence, this is the same as the "sharp" evaluation approach, except that some matches have been removed because less than half of the crowdsourcing group agreed with them. The continuous evaluation strategy penalises a matcher more if it misses a match on

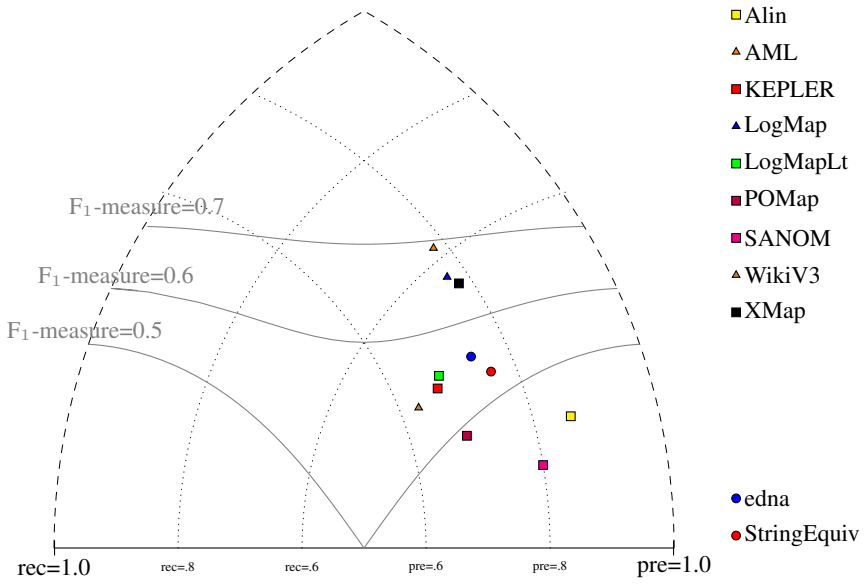


Fig. 2. Precision/recall triangular graph for the conference test case. Dotted lines depict level of precision/recall while values of F_1-measure are depicted by areas bordered by corresponding lines F_1-measure=0.[5|6|7].

which most people agree than if it misses a more controversial match. For instance, if $A \equiv B$ with a confidence of 0.85 in the reference alignment and a matcher gives that correspondence a confidence of 0.40, then that is counted as $0.85 \times 0.40 = 0.34$ true positive and $0.85 - 0.40 = 0.45$ false negative.

Out of the ten alignment matchers, three (ALIN, LogMapLt and ONTMAT) use 1.0 as the confidence value for all matches they identify. Two more have a narrow range of confidence values (POMap's values vary between 0.8 and 1.0, with the majority falling between 0.93 and 1.0 while SANOM's values are relatively tightly clustered between 0.73 and 0.9). The remaining five systems (AML, KEPLER, LogMap, WikiV3 and XMap) have a wide variation of confidence values.

When comparing the performance of the matchers on the uncertain reference alignments versus that on the sharp version (see Table 5), we see that in the discrete case all matchers performed the same or slightly better. Improvement in F-measure ranged from 0 to 8 percentage points over the sharp reference alignment. This was 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 most matchers is very similar regardless of whether a discrete or continuous evaluation methodology is used (provided that the threshold is optimized to achieve the highest possible F-measure in the discrete case). The primary exceptions to this are KEPLER, LogMap and SANOM. These systems perform significantly worse when evaluated using the continuous version of the metrics. In the LogMap and SANOM cases, this is because the matcher assigns low confidence values to some matches in which the labels are equivalent strings, which many crowdsourcers agreed with unless there was a compelling technical reason not to. This hurts recall, but using a low threshold value in the discrete version of the evaluation metrics 'hides' this problem. In the case of KEPLER, the issue is that entities whose labels share a word in common

Table 5. F-measure, precision, and recall of the different matchers when evaluated using the sharp (*ra1*), discrete uncertain and continuous uncertain metrics.

	Sharp			Discrete			Continuous		
Matcher	Prec.	F_1-m.	Rec.	Prec.	F_1-m.	Rec.	Prec.	F_1-m.	Rec.
ALIN	0.89	0.41	0.27	0.89	0.49	0.34	0.89	0.5	0.35
AML	0.84	0.74	0.66	0.79	0.78	0.77	0.8	0.77	0.74
KEPLER	0.76	0.59	0.48	0.76	0.67	0.6	0.58	0.62	0.68
LogMap	0.82	0.69	0.59	0.78	0.73	0.68	0.8	0.67	0.57
LogMapLt	0.73	0.59	0.5	0.72	0.67	0.62	0.72	0.67	0.63
ONTMAT	0.06	0.11	0.43	0.06	0.11	0.54	0.06	0.11	0.55
POMap	0.73	0.52	0.4	0.73	0.6	0.5	0.71	0.59	0.51
SANOM	0.81	0.38	0.25	0.81	0.45	0.31	0.81	0.38	0.25
WikiV3	0.67	0.57	0.49	0.74	0.62	0.52	0.73	0.63	0.55
XMap	0.84	0.68	0.57	0.79	0.72	0.67	0.81	0.73	0.67

have fairly high confidence values, even though they are often not equivalent. For example, "Review" and "Reviewing_Event". This hurts precision in the continuous case, but is taken care of by using a high threshold value in the discrete case.

Five matchers from this year also participated last year, and thus we are able to make some comparisons over time. The F-measures of all systems essentially held constant (within one percent) when evaluated against the uncertain reference alignments. This is in contrast to last year, in which most matchers made modest gains (in the neighborhood of 1 to 6 percent) over 2015. It seems that, barring any new advances, participating matchers have reached something of a steady state on this performance metric.

Evaluation based on violations of consistency and conservativity principles We performed evaluation based on detection of conservativity and consistency violations [44, 45]. The consistency principle states that correspondences should not lead to unsatisfiable classes in the merged ontology; the conservativity principle states that correspondences should not introduce new semantic relationships between concepts from one of the input ontologies.

Table 4 shows the number of unsatisfiable TBoxes after the ontologies are merged (Inc. Align.), the total number of all conservativity principle violations within all alignments (Conser.V.) and the total number of all consistency principle violations (Consist.V.).

Five tools (ALIN, AML, LogMap, ONTMAT and POMap) have no consistency principle violation (in comparison to seven last year) and two tools (SANOM and XMap) generated only one incoherent alignment. There is one tool (ALIN) having no conservativity principle violations. Further two tools (ONTMAT and POMap) have an average of conservativity principle violations around 1. 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.

4.3 Conclusions

In conclusion, this year four of ten matchers performed better than both baselines on sharp reference alignments. Further, this year five matchers generated coherent alignments (against seven matchers last year and five matchers the year before). Based on the uncertain reference alignments we can conclude that all matchers perform better on the fuzzy versus sharp version of the

benchmark and eight matchers have close correspondence on the continuous and discrete version, indicating good agreement with the human matchers. Finally, none of the five matchers that also participated last year improved their performance with regard to the evaluation based on the sharp or the uncertain reference alignments.

5 Large biomedical ontologies (largebio)

The largebio test case aims at finding alignments between the large and semantically rich biomedical ontologies FMA, SNOMED-CT, and NCI, which contain 78,989, 306,591 and 66,724 classes, respectively.

5.1 Test data

The test case has been split into three matching problems: FMA-NCI, FMA-SNOMED and SNOMED-NCI. Each matching problem has been further divided in 2 tasks involving differently sized fragments of the input ontologies: small overlapping fragments versus whole ontologies (FMA and NCI) or large fragments (SNOMED-CT).

The UMLS Metathesaurus [6] has been selected as the basis for reference alignments. UMLS is currently the most comprehensive effort for integrating independently-developed medical thesauri and ontologies, including FMA, SNOMED-CT, and NCI. The extraction of mapping from UMLS is detailed in [26]).

Since alignment coherence is an aspect of ontology matching that we aim to promote, in previous editions we provided coherent reference alignments by refining the UMLS mappings using the Alcomo (alignment) debugging system [32], LogMap's (alignment) repair facility [25], or both [27].

However, concerns were raised about the validity and fairness of applying automated alignment repair techniques to make reference alignments coherent [37]. It is clear that using the original (incoherent) UMLS alignments would be penalizing to ontology matching systems that perform alignment repair. However, using automatically repaired alignments would penalize systems that do not perform alignment repair and also systems that employ a repair strategy that differs from that used on the reference alignments [37].

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

To ensure that this solution was as fair as possible to all alignment repair strategies, we flagged as unknown all correspondences suppressed by any of Alcomo, LogMap or AML [39], as well as all correspondences suppressed from the reference alignments of last year's edition (using Alcomo and LogMap combined). Note that, we have used the (incomplete) repair modules of the above mentioned systems.

The flagged UMLS-based reference alignment for the OAEI 2017 campaign is summarized in Table 6.

Table 6. 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

5.2 Evaluation setting, participation and success

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

In the OAEI 2017 largebio track 10 out of 21 participating systems have been able to cope with at least one of the tasks of the largebio track with a 4 hours timeout. Note that we also include the results of *Tool1* (the developers withdrew the system from the campaign) as reference. 9 systems were able to complete more than one task, while 6 systems were able to complete all tasks. This is an improvement with respect to last year results where only 4 systems were able to complete all tasks

5.3 Background knowledge

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

LogMap uses normalisations and spelling variants from the general (biomedical) purpose UMLS Lexicon (a different resource with respect to the UMLS Metathesaurus).

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).

YAM-BIO uses as background knowledge a file containing mappings from the DOID and UBERON ontologies to other ontologies like FMA, NCI or SNOMED CT.

XMAP uses synonyms provided by the UMLS Metathesaurus. Note that matching systems using UMLS Metathesaurus as background knowledge will have a **notable advantage** since the largebio reference alignment is also based on the UMLS Metathesaurus.

5.4 Alignment coherence

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

We have used the OWL 2 reasoner HermiT [35] to compute the number of unsatisfiable classes. For the cases in which HermiT could not cope with the input ontologies and the alignments (in less than 2 hours) we have provided a lower bound on the number of unsatisfiable classes (indicated by $\geq$) using the OWL 2 EL reasoner ELK [28].

Table 7. System runtimes (in seconds) and task completion.

System	FMA-NCI		FMA-SNOMED		SNOMED-NCI		Average	#
	Task 1	Task 2	Task 3	Task 4	Task 5	Task 6		
LogMapLite	1	10	2	18	9	22	10	6
AML	44	77	109	177	669	312	231	6
LogMap	12	92	57	477	207	652	250	6
XMap	20	130	62	625	106	563	251	6
YAM-BIO	56	279	60	468	2,202	490	593	6
Tool1	65	1,650	245	2,140	481	1,150	955	6
LogMapBio	1,098	1,552	1,223	2,951	2,779	4,728	2,389	6
POMAP	595	-	1,841	-	-	-	1,218	2
SANOM	679	-	3,123	-	-	-	1,901	2
KEPLER	601	-	3,378	-	-	-	1,990	2
Wiki2	108,953	-	-	-	-	-	108,953	1
# Systems	**11**	**10**	**7**	**7**	**7**	**7**	**10,795**	**49**

In this OAEI edition, 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 [32]). 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).

Tables 8-9 (see last two columns) show that 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 [32], the repair module of LogMap (LogMap-Repair) [25] or the repair module of AML [39], which have worked well in practice [27, 23].

5.5 Runtimes and task completion

Table 7 shows which systems were able to complete each of the matching tasks in less than 4 hours and the required computation times. Systems have been ordered with respect to the number of completed tasks and the average time required to complete them. Times are reported in seconds.

The last column reports the number of tasks that a system could complete. For example, 7 system (including the withdrawn system *Tool1*) were able to complete all six tasks. The last row shows the number of systems that could finish each of the tasks. The tasks involving SNOMED were also harder with respect to both computation times and the number of systems that completed the tasks.

5.6 Results for the FMA-NCI matching problem

Table 8 summarizes the results for the tasks in the FMA-NCI matching problem.

XMap and YAM-BIO achieved the highest F-measure in Task 1, while XMap and AML in Task 2. Note however that the use of background knowledge based on the UMLS Metathesaurus has an important impact in the performance of XMap. The use of background knowledge led to

Table 8. Results for the FMA-NCI matching problem.

System	Time (s)	# Corresp.	Scores			Incoherence	
			Prec.	F-m.	Rec.	Unsat.	Degree
Task 1: small FMA and NCI fragments							
XMap*	20	2,649	0.98	0.94	0.90	2	0.019%
YAM-BIO	56	2,681	0.97	0.93	0.90	800	7.8%
AML	44	2,723	0.96	0.93	0.91	2	0.019%
LogMapBio	1,098	2,807	0.93	0.92	0.91	2	0.019%
LogMap	12	2,747	0.94	0.92	0.90	2	0.019%
KEPLER	601	2,506	0.96	0.89	0.83	3,707	36.1%
Average	10,193	2,550	0.95	0.89	0.84	1,238	12.0%
LogMapLite	1	2,483	0.97	0.89	0.82	2,045	19.9%
SANOM	679	2,457	0.95	0.87	0.80	1,183	11.5%
POMAP	595	2,475	0.90	0.86	0.83	3,493	34.0%
Tool1	65	2,316	0.97	0.86	0.77	1,128	11.0%
Wiki2	108,953	2,210	0.88	0.80	0.73	1,261	12.3%
Task 2: whole FMA and NCI ontologies							
XMap*	130	2,735	0.88	0.87	0.85	9	0.006%
AML	77	2,968	0.84	0.86	0.87	10	0.007%
YAM-BIO	279	3,109	0.82	0.85	0.89	11,770	8.1%
LogMap	92	2,701	0.86	0.83	0.81	9	0.006%
LogMapBio	1,552	2,913	0.82	0.83	0.83	9	0.006%
Average	541	2,994	0.80	0.81	0.83	7,389	5.1%
LogMapLite	10	3,477	0.67	0.74	0.82	26,478	18.1%
Tool1	1,650	3,056	0.69	0.71	0.74	13,442	9.2%

*Uses background knowledge based on the UMLS Metathesaurus which is the basis of the large-bio reference alignments.

an improvement in recall from LogMap-Bio over LogMap in both tasks, but this came at the cost of precision, resulting in the two variants of the system having identical F-measures.

Note that the effectiveness of the systems decreased from Task 1 to Task 2. One reason for this is that with larger ontologies there are more plausible mapping candidates, and thus it is harder to attain both a high precision and a high recall. Another reason is that the very scale of the problem constrains the matching strategies that systems can employ: AML for example, foregoes its matching algorithms that are computationally more complex when handling very large ontologies, due to efficiency concerns.

The size of Task 2 prove a problem for a number of systems, which were unable to complete it within the allotted time: POMAP, SANOM, KEPLER and Wiki2.

5.7 Results for the FMA-SNOMED matching problem

Table 9 summarizes the results for the tasks in the FMA-SNOMED matching problem.

XMap produced the best results in terms of both Recall and F-measure in Task 3 and Task 4, but again, we must highlight that it uses background knowledge based on the UMLS Metathesaurus. Among the other systems, AML and YAM-BIO achieved the highest F-measure in Tasks 3 and 4, respectively.

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

System	Time (s)	# Corresp.	Scores			Incoherence	
			Prec.	F-m.	Rec.	Unsat.	Degree
Task 3: small FMA and SNOMED fragments							
XMap*	62	7,400	0.97	0.91	0.85	0	0.0%
AML	109	6,988	0.92	0.84	0.76	0	0.0%
YAM-BIO	60	6,817	0.97	0.83	0.73	13,240	56.1%
LogMapBio	1,223	6,315	0.95	0.80	0.69	1	0.004%
LogMap	57	6,282	0.95	0.80	0.69	1	0.004%
Average	1,010	4,623	0.89	0.62	0.51	2,141	9.1%
KEPLER	3,378	4,005	0.82	0.56	0.42	3,335	14.1%
SANOM	3,123	3,146	0.69	0.42	0.30	2,768	11.7%
POMAP	1,841	2,655	0.68	0.42	0.30	1,013	4.3%
LogMapLite	2	1,644	0.97	0.34	0.21	771	3.3%
Tool1	245	979	0.99	0.24	0.14	287	1.2%
Task 4: whole FMA ontology with SNOMED large fragment							
XMap*	625	8,665	0.77	0.81	0.84	0	0.0%
YAM-BIO	468	7,171	0.89	0.80	0.73	54,081	26.8%
AML	177	6,571	0.88	0.77	0.69	0	0.0%
LogMap	477	6,394	0.84	0.73	0.65	0	0.0%
LogMapBio	2,951	6,634	0.81	0.72	0.65	0	0.0%
Average	979	5,470	0.84	0.63	0.56	8,445	4.2%
LogMapLite	18	1,822	0.85	0.34	0.21	4,389	2.2%
Tool1	2,140	1,038	0.87	0.23	0.13	649	0.3%

*Uses background knowledge based on the UMLS Metathesaurus which is the basis of the large-bio reference alignments.

Overall, the quality of the results was lower than that observed in the FMA-NCI matching problem, as the matching problem is considerable larger. Like in the FMA-NCI matching problem, the effectiveness off all systems decreases as the ontology size increases from Task 3 to Task 4; and of the systems that completed the former, for example, POMAP was unable to complete the latter.

5.8 Results for the SNOMED-NCI matching problem

Table 10 summarizes the results for the tasks in the SNOMED-NCI matching problem.

AML achieved the best results in terms of both Recall and F-measure in Tasks 5 and 6, while LogMap and AML achieved the best results in terms of precision in Tasks 5 and 6, respectively.

The overall performance of the systems was lower than in the FMA-SNOMED case, as this test case is even larger. Indeed, several systems were unable to complete even the smaller Task 5 within the allotted time: POMAP, SANOM and KEPLER.

As in the previous matching problems, effectiveness decreased as the ontology size increases. Unlike in the FMA-NCI and FMA-SNOMED matching problems, the use of the UMLS Metathesaurus did not positively impact the performance of XMap, which obtained lower results than expected.

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

System	Time (s)	# Corresp.	Scores			Incoherence	
			Prec.	F-m.	Rec.	Unsat.	Degree
Task 5: small SNOMED and NCI fragments							
AML	669	14,740	0.87	0.80	0.75	≥3,966	≥5.3%
LogMap	207	12,414	0.95	0.80	0.69	≥0	≥0.0%
LogMapBio	2,779	13,205	0.89	0.77	0.68	≥0	≥0.0%
YAM-BIO	2,202	12,959	0.90	0.77	0.68	≥549	≥0.7%
Average	921	12,220	0.89	0.70	0.59	21,264	28.3%
XMap*	106	16,968	0.89	0.69	0.57	≥46,091	≥61.3%
LogMapLite	9	10,942	0.89	0.69	0.57	≥60,450	≥80.4%
Tool1	481	4,312	0.87	0.35	0.22	≥37,797	≥50.2%
Task 6: whole NCI ontology with SNOMED large fragment							
AML	312	13,176	0.90	0.77	0.67	≥720	≥0.4%
YAM-BIO	490	15,027	0.83	0.76	0.70	≥2,212	≥1.2%
LogMapBio	4,728	13,677	0.84	0.73	0.64	≥5	≥0.003%
LogMap	652	12,273	0.87	0.71	0.60	≥3	≥0.002%
LogMapLite	22	12,894	0.80	0.66	0.57	≥150,656	≥79.5%
Average	1,131	13,666	0.84	0.66	0.56	55,496	29.3%
XMap*	563	23,707	0.82	0.66	0.55	≥137,136	≥72.4%
Tool1	1,150	4,911	0.81	0.34	0.22	≥97,743	≥51.6%

*Uses background knowledge based on the UMLS Metathesaurus which is the basis of the large-bio reference alignments.

6 Disease and Phenotype Track (phenotype)

The Pistoia Alliance Ontologies Mapping project team[9] organises this track based on a real use case where it is required to find alignments between disease and phenotype ontologies. Specifically, in the OAEI 2017 edition of this track the selected ontologies are the Human Phenotype Ontology (HPO), the Mammalian Phenotype Ontology (MP), the Human Disease Ontology (DOID), the Orphanet and Rare Diseases Ontology (ORDO), the Medical Subject Headings (MESH) ontology, and the Online Mendelian Inheritance in Man (OMIM) ontology. The extended results for the OAEI 2016 Disease and Phenotype track (previous campaign) are available in [24].

6.1 Test data

The 2017 edition comprises of four tasks requiring the pairwise alignment of:

- Human Phenotype Ontology (HP) to Mammalian Phenotype Ontology (MP);
- Human Disease Ontology (DOID) to the Orphanet Rare Disease Ontology (ORDO);
- Human Phenotype Ontology (HP) to Medical Subject Headings (MESH); and
- Human Phenotype Ontology (HP) to Online Mendelian Inheritance in Man (OMIM).

Currently, mappings 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.

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

Table 11. 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
MESH	Hoehndorf's version (2014)	BioPortal
OMIM	UMLS 2016AB	BioPortal

Table 11 summarizes the ontology versions and sources of the ontologies used in the OAEI 2017. Note that the version and source of HP, MP, DOID and ORDO are different from the ones used in 2016.

We have extracted "baseline" reference alignments based on the available BioPortal mappings (July 8, 2017). Most of the BioPortal [38] mappings are generated automatically by the LOOM[10] system, which should only be considered as a baseline since it is incomplete or may contain errors.

6.2 Evaluation setting

We have run the evaluation in a Ubuntu Laptop with an Intel Core i7-4600U CPU @ 2.10GHz x 4 and allocating 15Gb of RAM.

In the OAEI 2017 phenotype track 10 out of 21 participating OAEI 2017 systems have been able to cope with at least one of the tasks with 4 hours.

6.3 Evaluation criteria

Systems have been evaluated according to the following criteria:

- Precision and recall with respect to a consensus alignment automatically generated by voting based on the outputs of all participating systems (we have used vote=2, vote=3 and vote=4).
- Semantic recall with respect to manually generated mappings for several areas of interest (e.g., carbohydrate, obesity and breast cancer).
- Manual assessment of a subset unique mappings (i.e., mappings that are not suggested by other systems).

We have used the OWL 2 reasoner HermiT to calculate the semantic recall. For example, a positive hit will mean that a mapping in the reference has been (explicitly) included in the output mappings or it can be inferred using reasoning from the input ontologies and the output mappings.[11].

6.4 Use of background knowledge

LogMapBio uses BioPortal as mediating ontology provider, that is, it retrieves from BioPortal the most suitable top-10 ontologies for the matching task.

[10] `https://www.bioontology.org/wiki/index.php/BioPortal_Mappings`

[11] Details about the used notion of semantic precision and recall can be found in [24]

Table 12. Disease and Phenotype task completion.

System	HP-MP	DOID-ORDO	HP-MESH	HP-OMIM
AML	✓	✓	✓	✓
DiSMatch	✓	✓	✓	✓
LogMap	✓	✓	✓	✓
LogMapBio	✓	✓	✓	✓
LogMapLite	✓	✓	✓	*empty*
KEPLER	*time*	✓	*time*	*time*
POMAP	✓	✓	*time*	*time*
Tool1	✓	✓	✓	*empty*
XMap	✓	✓	✓	*empty*
YAM-BIO	✓	✓	✓	*empty*

✓: completed; *empty*: produced empty alignment; *error*: runtime error; *time*: timed out (4 hours).

LogMap uses normalisations and spelling variants from the general (biomedical) purpose UMLS Lexicon (a different resource with respect to the UMLS Metathesaurus).

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). Additionally, for the HPO-MP test case, it uses the logical definitions of both ontologies, which define some of their classes as being a combination of an anatomic term (i.e., a class from either FMA or Uberon) with a phenotype modifier term (i.e., a class from the Phenotypic Quality Ontology).

YAM-BIO uses as background knowledge a file containing mappings from the DOID and UBERON ontologies to other ontologies like FMA, NCI or SNOMED CT.

DiSMatch estimates the similarity among concepts through textual semantic relatedness. DiSMatch relies on a corpus of relevant biomedical textual resources.

XMAP uses synonyms provided by the UMLS Metathesaurus.

6.5 Results

AML, DiSMatch, LogMap, and LogMapBio produced the most complete results according to both the automatic and manual evaluation.

Table 12 summarizes the tasks where each system was able to produce results within a 4-hours time frame.

Results against the consensus alignments Table 13 shows the size of the consensus alignments built with the outputs of the systems participating in the OAEI 2017 campaign. Note that systems participating with different variants only contributed once in the voting, that is, the voting was done by family of systems/variants rather than by individual systems.

Table 3 shows the results achieved by each of the participating systems. We deliberately did not rank the systems since the consensus alignments only allow us to assess how systems perform in comparison with one another. On the one hand, some of the mappings in the consensus alignment may be erroneous (false positives), as all it takes for that is that 2, 3 or 4 systems agree on part of the erroneous mappings they find. On the other hand, the consensus alignments are not complete, as there will likely be correct mappings that no system is able to find, and as we will show in the manual evaluation, there are a number of mappings found by only one system

Table 13. Size of consensus alignments

Task	Vote 2	Vote 3	Vote 4
HP-MP	3,130	2,153	1,780
DOID-ORDO	3,354	2,645	2,188
HP-MESH	4,711	3,847	3,227
HP-OMIM	6,834	4,177	3,462

OM algorithm	Track Task	Total Equivalence Mappings	Precision Silver 2 Equiv mappings	Recall Silver 2 Equiv mappings	F-Score Silver 2 Equiv mappings	Sum F Scores Silver 2 Equiv mappings	Precision Silver 3 Equiv mappings	Recall Silver 3 Equiv mappings	F-Score Silver 3 Equiv mappings	Sum F Scores Silver 3 Equiv mappings	Precision Silver 4 Equiv mappings	Recall Silver 4 Equiv mappings	F-Score Silver 4 Equiv mappings	Sum F Scores Silver 4 Equiv mappings	Unique Equivalence Mappings	Precision for Unique Mappings
AML	HP-MP	2029	0.909	0.838	0.872	3.543	0.822	0.951	0.882	3.791	0.716	0.983	0.828	3.765	62	0.9333
AML	DOID-ORDO	4779	0.542	0.661	0.847		0.475	0.626	0.919		0.378	0.539	0.938		1520	0.8333
AML	HP-MESH	5638	0.799	0.871	0.957		0.677	0.805	0.992		0.572	0.727	0.999		678	
AML	HP-OMIM	6681	0.888	0.878	0.868		0.624	0.768	0.998		0.518	0.683	1.000		679	
BioPortal	HP-MP	696	0.999	0.316	0.480	1.955	0.999	0.396	0.567	2.599	0.996	0.469	0.638	3.034	1	
BioPortal	DOID-ORDO	1237	0.998	0.575	0.403		0.998	0.666	0.500		0.996	0.779	0.639		2	
BioPortal	HP-MESH	2466	0.998	0.686	0.523		0.994	0.776	0.637		0.990	0.858	0.757		1	
BioPortal	HP-OMIM	3768	0.995	0.708	0.549		0.992	0.941	0.895		0.919	0.958	1.000		16	
DiSMatch AR	HP-MP	2378	0.592	0.640	0.615	2.755	0.500	0.678	0.576	3.144	0.453	0.729	0.559	3.330	831	0.8333
DiSMatch AR	DOID-ORDO	3130	0.591	0.598	0.604		0.539	0.603	0.684		0.504	0.624	0.818		1234	0.6333
DiSMatch AR	HP-MESH	9161	0.428	0.565	0.833		0.385	0.542	0.917		0.342	0.506	0.971		4928	
DiSMatch AR	HP-OMIM	7356	0.653	0.677	0.703		0.549	0.701	0.967		0.462	0.628	0.982		2495	
DiSMatch SG	HP-MP	2318	0.618	0.651	0.634	2.207	0.520	0.688	0.592	2.500	0.469	0.735	0.572	2.543	771	0.8000
DiSMatch SG	DOID-ORDO	0													0	
DiSMatch SG	HP-MESH	9072	0.434	0.571	0.835		0.390	0.548	0.920		0.347	0.511	0.974		4830	
DiSMatch SG	HP-OMIM	7668	0.658	0.695	0.738		0.538	0.696	0.987		0.450	0.620	0.997		2572	
DiSMatch TR	HP-MP	2331	0.614	0.650	0.632	2.814	0.517	0.687	0.590	3.183	0.465	0.733	0.569	3.361	782	0.8333
DiSMatch TR	DOID-ORDO	3089	0.600	0.602	0.605		0.545	0.606	0.682		0.510	0.628	0.817		1192	0.6333
DiSMatch TR	HP-MESH	9138	0.433	0.571	0.839		0.389	0.547	0.924		0.345	0.510	0.978		4885	
DiSMatch TR	HP-OMIM	7680	0.657	0.695	0.738		0.537	0.696	0.988		0.450	0.620	0.997		2575	
KEPLER	HP-MP	0													0	
KEPLER	DOID-ORDO	1824	0.919	0.686	0.547	0.547	0.860	0.730	0.635	0.635	0.812	0.789	0.768	0.768	131	0.8667
KEPLER	HP-MESH	0														
KEPLER	HP-OMIM	0														
LogMap	HP-MP	2124	0.876	0.845	0.860	2.991	0.767	0.929	0.840	3.149	0.676	0.972	0.798	3.240	189	0.9330
LogMap	DOID-ORDO	2396	0.981	0.861	0.768		0.903	0.890	0.876		0.744	0.824	0.924		41	0.6667
LogMap	HP-MESH	2291	0.938	0.614	0.456		0.869	0.649	0.518		0.766	0.636	0.544		82	
LogMap	HP-OMIM	7202	0.860	0.883	0.906		0.531	0.672	0.915		0.468	0.632	0.974		984	
LogMapBio	HP-MP	2204	0.859	0.860	0.860	3.176	0.749	0.941	0.834	3.291	0.656	0.978	0.785	3.366	218	0.9333
LogMapBio	DOID-ORDO	2620	0.933	0.861	0.798		0.845	0.871	0.897		0.692	0.798	0.941		136	0.7667
LogMapBio	HP-MESH	2948	0.908	0.699	0.568		0.810	0.703	0.621		0.701	0.669	0.640		158	
LogMapBio	HP-OMIM	7725	0.840	0.891	0.950		0.508	0.659	0.939		0.448	0.619	0.999		1174	
LogMapLite	HP-MP	725	0.997	0.329	0.494	1.541	0.997	0.412	0.583	1.866	0.997	0.490	0.657	2.199	0	
LogMapLite	DOID-ORDO	1251	0.995	0.577	0.407		0.994	0.669	0.504		0.994	0.782	0.645		0	
LogMapLite	HP-MESH	3017	0.999	0.780	0.640		0.994	0.874	0.779		0.960	0.928	0.897		2	
LogMapLite	HP-OMIM	0														
Tool1	HP-MP	1530	0.921	0.640	0.755	1.951	0.895	0.781	0.834	2.248	0.830	0.860	0.845	2.479	31	0.8000
Tool1	DOID-ORDO	1711	0.996	0.714	0.556		0.981	0.803	0.680		0.943	0.887	0.837		7	1.0000
Tool1	HP-MESH	3057	0.984	0.774	0.639		0.923	0.817	0.734		0.841	0.818	0.797		10	
Tool1	HP-OMIM	0														
POMAP	HP-MP	2024	0.764	0.703	0.732	1.558	0.687	0.793	0.736	1.639	0.615	0.843	0.711	1.636	402	0.7000
POMAP	DOID-ORDO	3222	0.785	0.805	0.826		0.691	0.783	0.902		0.553	0.692	0.925		666	0.2333
POMAP	HP-MESH	0														
POMAP	HP-OMIM	0														
XMap	HP-MP	1201	0.981	0.535	0.693	1.808	0.960	0.657	0.780	2.120	0.942	0.766	0.845	2.444	15	0.9286
XMap	DOID-ORDO	1587	0.971	0.663	0.503		0.959	0.750	0.616		0.931	0.841	0.767		41	0.6667
XMap	HP-MESH	2955	0.975	0.752	0.612		0.942	0.819	0.724		0.909	0.869	0.833		59	
XMap	HP-OMIM	0														
YAM-BIO	HP-MP	883	1.000	0.401	0.573	1.550	0.999	0.503	0.669	1.860	0.984	0.588	0.736	2.160	0	
YAM-BIO	DOID-ORDO	1315	0.996	0.599	0.428		0.992	0.690	0.529		0.989	0.802	0.674		5	1.0000
YAM-BIO	HP-MESH	2610	0.992	0.707	0.550		0.977	0.790	0.663		0.927	0.829	0.750		21	
YAM-BIO	HP-OMIM	0														

Fig. 3. Results against consensus alignments with vote 2, 3 and 4.

(and therefore not in the consensus alignments) which are correct. Nevertheless, the results with respect to the consensus alignments do provide some insights into the performance of the systems, which is why we highlighted in the table the 4 systems that produce results closest to the silver standards: AML, DiSMatch, LogMap, and LogMapBio.

Results against manually created mappings The manually generated mappings for six areas (carbohydrate, obesity and breast cancer, urinary incontinence, abnormal heart and Charcot-Marie Tooth disease) include 86 mappings between HP and MP and 175 mappings between DOID and ORDO. Most of them represent subsumption relationships. Tables 14 and 15 shows the results in terms of recall and semantic recall for each of the system. LogMapBio and LogMap

Table 14. Results against manually created mappings: HP-MP task

System	Standard Recall	Semantic Recall
BioPortal (baseline)	*0.20*	*0.51*
AML	0.40	0.62
DiSMatch-ar	0.42	0.65
LogMap	0.38	**0.67**
LogMapBio	0.38	**0.67**
LogMapLt	0.20	0.51
Tool1	0.31	0.60
POMap	0.38	0.65
XMap	0.30	0.60
YAM-BIO	0.22	0.51

Table 15. Results against manually created mappings: DOID-ORDO task

System	Standard Recall	Semantic Recall
BioPortal (baseline)	*0.13*	*0.14*
AML	0.33	**0.48**
DiSMatch-ar	0.21	0.25
DiSMatch-sg	0.21	0.25
DiSMatch-tr	0.21	0.25
KEPLER	0.13	0.17
LogMap	0.30	0.42
LogMapBio	0.32	0.44
LogMapLt	0.13	0.14
Tool1	0.27	0.30
POMap	0.27	0.30
XMap	0.13	0.14
YAM-BIO	0.13	0.14

obtain the best results in terms of semantic recall in the HP-MP task, while AML obtains the best results in the DOID-ORDO task. The results in both tasks are far from optimal since a large fragment of the manually created mappings have not been (explicitly) identified by the systems nor can be derived via reasoning.

Manual assessment of unique mappings Figures 4 and 5 show the results of the manual assessment to estimate the precision of the unique mappings generated by the participating systems. Unique mappings are correspondences that no other system (explicitly) provided in the output. We manually evaluated up to 30 mappings and we focused the assessment on unique equivalence mappings.

For example LogMap's output contains 189 unique mappings in the HP-MP task. The manual assessment revealed an (estimated) precision of 0.9333. In order to also take into account the number of unique mappings that a system is able to discover, Tables 4 and 5 also include the estimation of the positive and negative contribution of the unique mappings with respect to the total unique mappings discovered by all participating systems.

System	Task	Unique Mappings	Precision	Positive contribution ratio (%)	Negative contribution ratio (%)
AML	HP-MP	62	0.9333	1.75%	0.13%
DiSMatch AR	HP-MP	831	0.8333	21.00%	4.20%
DiSMatch SG	HP-MP	771	0.8000	18.70%	4.68%
DiSMatch TR	HP-MP	782	0.8333	19.76%	3.95%
KEPLER	HP-MP	0			
LogMap	HP-MP	189	0.9330	5.35%	0.38%
LogMapBio	HP-MP	216	0.9333	6.11%	0.44%
LogMapLite	HP-MP	0			
Tool1	HP-MP	31	0.8000	0.75%	0.19%
POMAP	HP-MP	402	0.7000	8.53%	3.66%
XMap	HP-MP	14	0.9286	0.39%	0.03%
YAM-BIO	HP-MP	0			
Totals		**3298**		**82.35%**	**17.65%**

Fig. 4. Unique mappings in the HP-MP task.

System	Task	Unique Mappings	Precision	Positive contribution ratio (%)	Negative contribution ratio (%)
AML	DOID-ORDO	1520	0.8333	25.48%	5.10%
DiSMatch AR	DOID-ORDO	1234	0.6333	15.72%	9.10%
DiSMatch SG	DOID-ORDO	0			
DiSMatch TR	DOID-ORDO	1192	0.6333	15.19%	8.79%
KEPLER	DOID-ORDO	131	0.8667	2.28%	0.35%
LogMap	DOID-ORDO	40	0.6667	0.54%	0.27%
LogMapBio	DOID-ORDO	135	0.7667	2.08%	0.63%
LogMapLite	DOID-ORDO	0			
Tool1	DOID-ORDO	7	1.0000	0.14%	0.00%
POMAP	DOID-ORDO	666	0.2333	3.13%	10.27%
XMap	DOID-ORDO	41	0.6667	0.55%	0.27%
YAM-BIO	DOID-ORDO	5	1.0000	0.10%	0.00%
Totals	DOID-ORDO	**4971**		65.21%	34.79%

Fig. 5. Unique mappings in the DOID-ORDO task.

7 MultiFarm

The MultiFarm data set [33] aims at evaluating the ability of matching systems to deal with ontologies in different natural languages. This data set results from the translation of 7 ontologies from the conference track (cmt, conference, confOf, iasted, sigkdd, ekaw and edas) into 10 languages: Arabic, Chinese, Czech, Dutch, French, German, Italian, Portuguese, Russian, and Spanish. It is composed of 55 pairs of languages (see [33] for details on how the original MultiFarm data set has been generated). For each pair, taking into account the alignment direction (cmt$_{en}$ →confOf$_{de}$ and cmt$_{de}$ →confOf$_{en}$, for instance, as distinct matching tasks), we have 49 matching tasks. The whole data set is composed of 55×49 matching tasks.

7.1 Experimental setting

Part of the data set is used for blind evaluation. This subset includes all matching tasks involving the *edas* and *ekaw* ontologies (resulting in 55×24 matching tasks). As last year, the results reported here are based on the blind data set. Participants were able to test their systems on the available subset of matching tasks (*open evaluation*), available via the SEALS repository. The open subset covers 45×25 tasks. The open subset does not include Italian translations.

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

This year, 8 systems (out of 22) have participated in the MultiFarm track (i.e., those that have been assigned to the task in the registration phase) : AML, CroLOM, KEPLER, LogMap, LogMapLite, SANOM, WikiV3, and XMAP. LogMapLite does not implement any specific cross-lingual strategy. The number of participants is stable with respect to the last campaign (7 in 2016, 5 in 2015, 3 in 2014, 7 in 2013, and 7 in 2012). For sake of simplicity, we refer in the following to cross-lingual systems those implementing cross-lingual matching strategies and non-cross-lingual systems those without that feature. The reader can refer to the OAEI papers for a detailed description of the strategies adopted by each system. In fact, most of them still adopts a translation step before the matching itself.

For this track, the general comments with respect to the running are : i) CroLOM participated with the same version than last year; ii) LogMap had encountered problems for accessing the Google translator server; iii) KEPLER generated some parsing errors for some pairs; iv) some systems (AML, LogMap and LogMapLite) have generated correspondences with confidence higher than 1.0 (no post-processing has been done in these cases).

7.2 Execution setting and runtime

The systems have been executed on a Windows machine configured with 8GB of RAM running under a i7-7500U CPU 2.70GHz x4 processors. All measurements are based on a single run. As Table 16 shows, we can observe large differences in the time required for a system to complete the 55 x 24 matching tasks. Note as well that the concurrent access to the SEALS repositories during the evaluation period may have an impact in the time required for completing the task.

7.3 Evaluation results

Table 16 presents the aggregated results for the 55×24 matching tasks. They have been computed using the Alignment API 4.6 and can slightly differ from those computed with the SEALS client. We haven't applied any threshold on the results. They are measured in terms of classical precision and recall.

Overall, as expected, systems implementing cross-lingual techniques outperform the non-cross-lingual systems. However, as stated above, this year we did not run all systems and focus on the systems that have been registered for the task. In this task, AML outperforms all other systems in terms of F-measure for task i), keeping its top place in this task. AML is followed by LogMap, CroLOM, KEPLER and WikiV3. With respect to the task ii), AML has relatively low performance, due mainly to some errors in parsing the alignments for which a confidence higher than 1 was generated. KEPLER has provided the higher F-measure for task ii), followed by LogMap, CroLOM and AML. We observe that WikiV3 is able to maintain its performance in both tasks.

Table 16. 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	677	55	8.21	.72(.72)	.46(.46)	.35(.35)	45.54	.89(.96)	.26(.28)	.16(.17)
CroLOM	5501	55	8.56	.55(.55)	.36(.36)	.28(.28)	38.76	.89(.90)	.40(.40)	.26(.27)
KEPLER	2180	55	10.63	.43(.43)	.31(.31)	.25(.25)	58.34	.90(.90)	.52(.52)	.38(.38)
LogMap	57	55	6.99	.73(.73)	.37(.37)	.25(.25)	46.80	.95(.96)	.42(.43)	.28(.28)
LogMapLite	38	55	1.16	.36(.36)	.04(.04)	.02(.02)	94.5	.02(.02)	.01(.03)	.01(.02)
SANOM	22	30	2.86	.43(.79)	.13(.25)	.08(.15)	8.33	.54(.99)	.06(.12)	.03(.06)
WikiV3	1343	55	11.89	.30(.30)	.25(.25)	.21(.21)	29.37	.62(.62)	.23(.23)	.14(.14)
XMAP	102	27	3.84	.24(.50)	.06(.14)	.04(.09)	15.76	.66(.91)	.10(.14)	.06(.09)

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 generated (non empty) alignments; size indicates the average of the number of generated correspondences for the tests where an (non empty) alignment has been generated. Two kinds of results are reported: those do not distinguishing empty and erroneous (or not generated) alignments and those—indicated between parenthesis—considering only non empty generated alignments for a pair of languages.

With respect to the pairs of languages for test cases of type i), for the sake of brevity, we do not present the results for the 55 pairs. The reader can refer to the OAEI results web page for the detailed results. 5 cross-lingual systems out of 7 were able to deal with all pairs of languages (AML, CroLOM, KEPLER, LogMap and WikiV3). While the only non-specific system was able to generate non empty (but erroneous) results for all pairs, specific systems as SANOM and XMap have problems to deal with ar, cn and ru languages and hence were not able to generate alignments for most pairs involving these languages. This behaviour has also been observed in the last campaign for specific systems.

For the group of systems implementing cross-lingual strategies, their top F-measure include the pairs es-it (AML), nl-pt (CroLOM), de-pt (KEPLER), en-nl (LogMap), es-it (SANOM), it-pt (WikiV3), es-pt (XMap). We can observe that most of the systems better deal with the pairs involving pt, it, es, nl, de and en languages. This may due to the coverage or performance of the resources and translations for these languages, together with the fact that dealing with comparable languages[12] can make the task easier. In fact, we can also observe that for most systems, the worst results have been produced for the pairs involving ar, cn, cz and ru. The exceptions are SANOM and XMap, for which, worst results also include the pairs es, nl and pt or fr, en and it, respectively.

With respect to the only non cross-lingual systems, LogMapLite, it in fact takes advantage of comparable languages, in the absence of specific strategies. This can be corroborated by the fact that it has generated its best F-measure for the pairs de-en, es-pt, it-pt, es-it. This (expected) fact has been observed along the campaigns.

[12] An example of comparable natural languages is English and German, both belonging to the Germanic language family. Comparable natural languages can also be languages that are not from the same language family. For example, Italian belonging to the Romance language family, and German belonging to the Germanic language family can still be compared using string comparison techniques such as edit distance, as they are both alphabetic letter-based with comparable graphemes. An example of natural languages that are not comparable in this context can be Chinese and English, where the former is logogram-based and the latter is alphabetic letter-based [12]

Comparison with previous campaigns. The number of participants implementing cross-lingual strategies remains stable this year with respect to the last campaigns (7 in 2016, 5 in 2015, 3 in 2014, 7 in 2013 and 2012 and 3 in 2011). 4 systems have also participated last year (AML, LogMap, CroLOM, and XMap) and we count 3 new systems (KEPLER, SANOM, and WikiV3). Comparing the results from last year, in terms F-measure and with respect to the blind evaluation (cases of type i), AML maintains its performance, with a very little increase (.46 in 2017, .45 in 2016 and .47 in 2015). CroLOM, LogMap, and XMAP maintained their performance (.36, .37 and .06, respectively). The newcomer WikiV3 obtained stable results for both kinds of tasks, but with a F-measure below AML, LogMap, CroLOM and KEPLER. For the task ii), we can observe that KEPLER (.52) outperforms LogMap (.44), the best system from last year, in terms of F-measure for this task.

7.4 Conclusion

From 22 participants, 8 were evaluated in MultiFarm. In terms of performance, the F-measure for blind tests remains relatively stable across campaigns. AML and LogMap keep their positions with respect to the previous campaigns, followed by the CroLOM and KEPLER. Still, all systems privilege precision in detriment to recall and the results are below the ones obtained for the Conference original dataset. We can observe as well that the systems are not able to provide good results or deal with pairs involving specific languages, as ar, cn and ru. As last years, still cross-lingual approaches are mainly based on translation strategies and the combination of other resources (like cross-lingual links in Wikipedia, BabelNet, etc.) and strategies (machine learning, indirect alignment composition) remains underexploited. As last year, the evaluation has been conducted only on the blind set (results have not been reported for the open data set). As future work, we plan to compare the performance of the systems on both multilingual and cross-lingual settings.

8 Interactive matching

The interactive matching track was organized at OAEI 2017 for the fifth time. The goal of this evaluation is to simulate interactive matching [36, 14], where a human expert is involved to validate correspondences found by the matching system. In the evaluation, we look at how interacting with the user improves the matching results. Currently, this track does not evaluate the user experience or the user interfaces of the systems.

8.1 Datasets

The Interactive track uses four OAEI datasets: Anatomy (Section 3), Conference (Section 4), LargeBio (Section 5), and Phenotype (Section 6). For details on the datasets, please refer to their respective sections.

8.2 Experimental setting

The Interactive track relies on the SEALS client's *Oracle* class to simulate user interactions. An interactive matching system can present a correspondence to the oracle, which will tell the system whether that correspondence is right or wrong. This year we have extended this functionality by allowing a user to present a collection of mappings simultaneously to the oracle. If a system presents up to three mappings together and each mapping presented has a mapped entity (i.e.,

class or property) in common with at least one other mapping 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 mappings.

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.

The evaluations of the Conference and Anatomy datasets were run on a server with 3.46 GHz (6 cores) and 8GB RAM allocated to the matching 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 represent the total number of interactions for all the pairs. Both are averaged for the ten runs.

The Phenotype and Largebio evaluation was run on a Ubuntu Laptop with an Intel Core i7-4600U CPU @ 2.10GHz x 4 and allocating 15Gb of RAM. Each system was run only one time due to the time required to run some of the systems. Since errors are randomly introduced we expect minor variations between runs. Nevertheless, the Phenotype and Largebio tasks involve large ontologies and a comparatively large number of questions, hence the variations between runs are expected to be mostly negligible.

8.3 Evaluation

For the sake of brevity, we present only the results for the Anatomy, Conference, and LargeBio tasks. For the Phenotype tasks, please refer to the OAEI website [13]. Table 17 and Figure 6 show the results for the Anatomy and Conference datasets, and Table 18 and Figure 7 show the results for the LargeBio tasks.

The tables include the following information (column names within parentheses):

- The number of unsatisfiable classes resulting from the alignments computed as detailed in Section 5 - only for the LargeBio data set.
- 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 (as detailed in Section 3). 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 mappings, that could be analysed simultaneously by a user.
- Distinct mappings (Dist. Mapps) counts the total number of mappings 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).

[13] http://oaei.ontologymatching.org/2017/results/interactive/

The figures show the time intervals between the questions to the user/oracle for the different systems and error rates. Different runs are depicted with different colours.

8.4 Discussion

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 mappings, ALIN can also add new candidate mappings to its initial set. LogMap and AML both request feedback on only selected mapping candidates (based on their similarity patterns or their involvement in unsatisfiabilities) and AML presents one mapping at a time to the user. XMap also presents one mapping at a time and asks mainly about false mappings. ALIN and LogMap can both ask the oracle to analyse several conflicting mappings simultaneously.

The performance of the systems usually improves when interacting with a perfect oracle in comparison with no interaction. The one exception is XMap in the Conference dataset, because it is barely interactive in this dataset. In general, XMap performs very few requests to the oracle compared to the other systems, except in the SNOMED-NCI task, where it makes the most requests. 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, not only because it makes a high number of oracle requests (the most in Anatomy and Conference) but also because its non-interactive performance was the lowest of the interactive systems, and thus the easiest to improve.

Although systems' 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 its 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 data sets, and for XMap in the SNOMED-NCI task, as the F-measure according to the oracle decreases as the error rate increases. This means that the latter systems are deliberately or implicitly letting the oracle's replies affect their selection of mappings beyond those they asked about, and thus propagating the oracle's errors.

Two models for system *response times* are frequently used in the literature [10]: Shneiderman and Seow take different approaches to categorise the response times. Shneiderman takes a task-centred view and sorts the response times in four categories according to task complexity: typing, mouse movement (50-150 ms), simple frequent tasks (1 s), common tasks (2-4 s) and complex tasks (8-12 s). He suggests that the user is more tolerable to delays with the growing complexity of the task at hand. Unfortunately, no clear definition is given for how to define the task complexity. Seow's model looks at the problem from a user-centred perspective by considering the user expectations towards the execution of a task: instantaneous (100-200 ms), immediate (0.5-1 s), continuous (2-5 s), captive (7-10 s). Ontology 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 data sets 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 data sets. 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).

Regarding the number of unsatisfiable classes resulting from the alignments we observe some expected variations as the error increases. We note that, with interaction, the alignments

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	0.985	0.339	0.504	0.0	–	–	–	–	–	–	–
	0.0	0.993	0.794	0.882	0.454	0.993	0.794	0.882	939	1472	1.0	1.0
	0.1	0.94	0.745	0.831	0.403	0.993	0.79	0.88	905	1352	0.905	0.8977
	0.2	0.895	0.703	0.787	0.358	0.993	0.788	0.879	891	1311	0.824	0.796
	0.3	0.846	0.649	0.735	0.301	0.993	0.781	0.874	882	1266	0.734	0.668
AML	NI	0.95	0.936	0.943	0.832	–	–	–	–	–	–	–
	0.0	0.968	0.948	0.958	0.862	0.968	0.948	0.958	241	240	1.0	1.0
	0.1	0.956	0.946	0.95	0.856	0.969	0.949	0.959	266	264	0.73	0.972
	0.2	0.939	0.942	0.94	0.849	0.969	0.951	0.96	283	280	0.513	0.93
	0.3	0.922	0.939	0.931	0.843	0.97	0.952	0.961	310	308	0.359	0.902
LogMap	NI	0.911	0.846	0.877	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	0.962	0.83	0.891	0.564	0.966	0.803	0.877	388	1164	0.748	0.964
	0.2	0.944	0.823	0.88	0.552	0.945	0.762	0.843	388	1164	0.566	0.927
	0.3	0.931	0.82	0.872	0.544	0.92	0.722	0.809	388	1164	0.431	0.879
XMap	NI	0.926	0.863	0.893	0.639	–	–	–	–	–	–	–
	0.0	0.927	0.865	0.895	0.644	0.927	0.865	0.895	35	35	1.0	1.0
	0.1	0.927	0.865	0.895	0.644	0.927	0.863	0.894	35	35	0.602	0.964
	0.2	0.927	0.865	0.895	0.644	0.927	0.862	0.893	35	35	0.422	0.964
	0.3	0.927	0.865	0.895	0.644	0.927	0.861	0.893	35	35	0.278	0.93
Conference Dataset												
ALIN	NI	0.892	0.272	0.417	–	–	–	–	–	–	–	–
	0.0	0.957	0.731	0.829	–	0.957	0.731	0.829	329	571	1.0	1.0
	0.1	0.804	0.669	0.73	–	0.961	0.737	0.834	321	549	0.752	0.966
	0.2	0.669	0.622	0.645	–	0.965	0.751	0.845	313	534	0.558	0.93
	0.3	0.577	0.56	0.568	–	0.966	0.752	0.845	302	517	0.431	0.875
AML	NI	0.841	0.659	0.739		–	–	–	–	–	–	–
	0.0	0.912	0.711	0.799	–	0.912	0.711	0.799	271	270	1.0	1.0
	0.1	0.841	0.701	0.765	–	0.923	0.732	0.816	282	275	0.704	0.975
	0.2	0.768	0.672	0.717	–	0.925	0.745	0.825	292	279	0.538	0.92
	0.3	0.713	0.651	0.68	–	0.929	0.751	0.83	291	274	0.45	0.877
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	0.851	0.598	0.702	–	0.855	0.573	0.686	82	246	0.698	0.978
	0.2	0.821	0.585	0.684	–	0.829	0.542	0.656	82	246	0.507	0.941
	0.3	0.795	0.581	0.671	–	0.807	0.518	0.631	82	246	0.363	0.902
XMap	NI	0.837	0.57	0.678	–	–	–	–	–	–	–	–
	0.0	0.837	0.57	0.678	–	0.837	0.57	0.678	4	4	0.0	1.0
	0.1	0.837	0.57	0.678	–	0.837	0.57	0.678	4	4	0.0	1.0
	0.2	0.837	0.57	0.678	–	0.837	0.569	0.677	4	4	0.0	1.0
	0.3	0.837	0.57	0.678	–	0.837	0.569	0.678	4	4	0.0	1.0

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

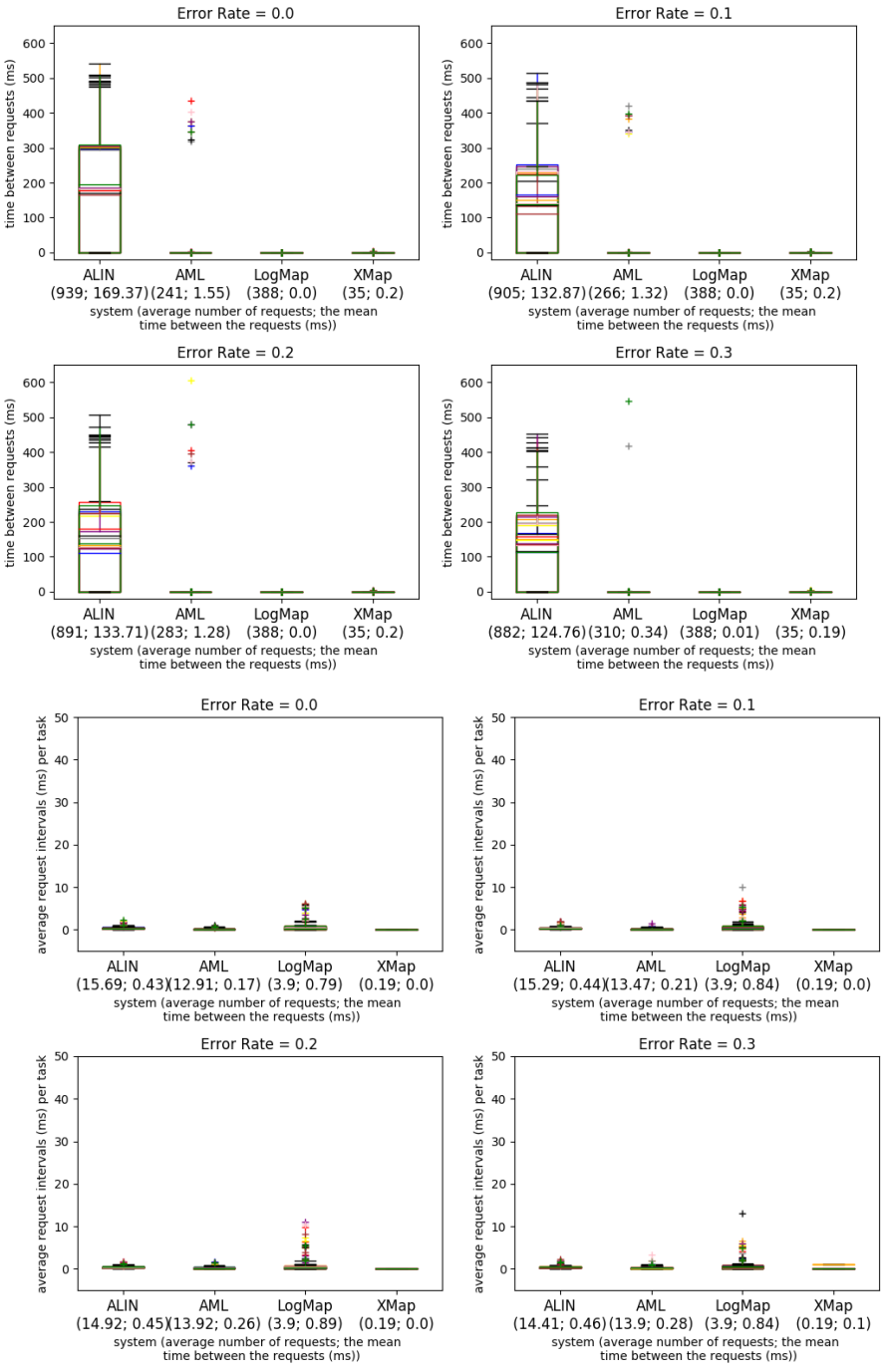


Fig. 6. Time intervals between requests to the user/oracle for the Anatomy (top 4 plots) and Conference (bottom 4 plots) datasets. Whiskers: Q1-1,5IQR, Q3+1,5IQR, IQR=Q3-Q1. The labels under the system names show the average number of requests and the mean time between the requests for the ten runs.

Table 18. Interactive matching results for the LargeBio dataset

Tool	Error	Unsat.	Prec.	Rec.	F-m.	Prec. oracle	Rec. oracle	F-m. oracle	Tot. Reqs.	Dist. Mapps	Pos. Prec.	Neg. Prec.
FMA-NCI Small Dataset												
ALIN	NI	N/A	0.995	0.455	0.624	–	–	–	–	–	–	–
	0.0	2	0.996	0.63	0.772	0.996	0.63	0.772	653	1,019	1	1
	0.1	85	0.971	0.614	0.752	0.996	0.63	0.772	629	932	0.908	0.907
	0.2	152	0.958	0.593	0.733	0.996	0.624	0.767	605	881	0.855	0.788
	0.3	91	0.937	0.58	0.716	0.996	0.623	0.767	589	855	0.772	0.696
AML	NI	2	0.963	0.902	0.932	–	–	–	–	–	–	–
	0.0	2	0.99	0.913	0.95	0.99	0.913	0.95	449	447	1	1
	0.1	222	0.98	0.908	0.943	0.99	0.914	0.95	497	484	0.896	0.936
	0.2	2	0.974	0.894	0.932	0.987	0.91	0.947	450	450	0.794	0.768
	0.3	2	0.966	0.894	0.929	0.981	0.911	0.945	450	450	0.751	0.734
LogMap	NI	2	0.944	0.897	0.92	–	–	–	–	–	–	–
	0.0	2	0.992	0.901	0.944	0.992	0.901	0.944	1,131	1,131	1	1
	0.1	2	0.98	0.881	0.928	0.983	0.892	0.935	1,209	1,209	0.942	0.909
	0.2	2	0.967	0.874	0.918	0.964	0.875	0.917	1,247	1,247	0.837	0.84
	0.3	2	0.963	0.872	0.915	0.935	0.849	0.89	1,327	1,327	0.727	0.776
XMap	NI	2	0.977	0.901	0.937	–	–	–	–	–	–	–
	0.0	2	0.991	0.9	0.943	0.991	0.9	0.943	188	188	1	1
	0.1	2	0.988	0.895	0.939	0.99	0.9	0.943	187	187	0.962	0.819
	0.2	2	0.988	0.892	0.938	0.99	0.899	0.942	187	187	0.939	0.753
	0.3	2	0.985	0.887	0.933	0.99	0.899	0.942	188	188	0.851	0.628
SNOMED-NCI Small Dataset												
AML	NI	3,966	0.904	0.713	0.797	–	–	–	–	–	–	–
	0.0	0	0.972	0.726	0.831	0.972	0.726	0.831	2,730	2,730	1	1
	0.1	0	0.967	0.717	0.823	0.972	0.724	0.83	2,730	2,730	0.942	0.857
	0.2	0	0.961	0.707	0.815	0.972	0.721	0.828	2,730	2,730	0.88	0.732
	0.3	0	0.955	0.697	0.806	0.972	0.719	0.827	2,730	2,730	0.818	0.622
LogMap	NI	0	0.922	0.663	0.771	–	–	–	–	–	–	–
	0.0	0	0.985	0.669	0.797	0.985	0.669	0.797	5,596	5,596	1	1
	0.1	16	0.974	0.651	0.78	0.971	0.656	0.783	6,201	6,201	0.945	0.855
	0.2	16	0.965	0.64	0.77	0.948	0.639	0.763	6,737	6,737	0.859	0.766
	0.3	16	0.959	0.635	0.764	0.92	0.62	0.741	7,159	7,159	0.753	0.693
XMap	NI	46,091	0.911	0.564	0.697	–	–	–	–	–	–	–
	0.0	35,869	0.924	0.59	0.72	0.924	0.59	0.72	11,932	11,689	1	1
	0.1	35,455	0.923	0.591	0.721	0.84	0.568	0.678	11,931	11,694	0.99	0.602
	0.2	35,968	0.921	0.591	0.72	0.754	0.541	0.63	11,911	11,682	0.975	0.41
	0.3	36,619	0.919	0.592	0.72	0.676	0.514	0.584	11,903	11,693	0.953	0.297

NI stands for non-interactive, and refers to the results obtained by the matching system in the original track. ALIN was unable to complete the SNOMED-NCI task.

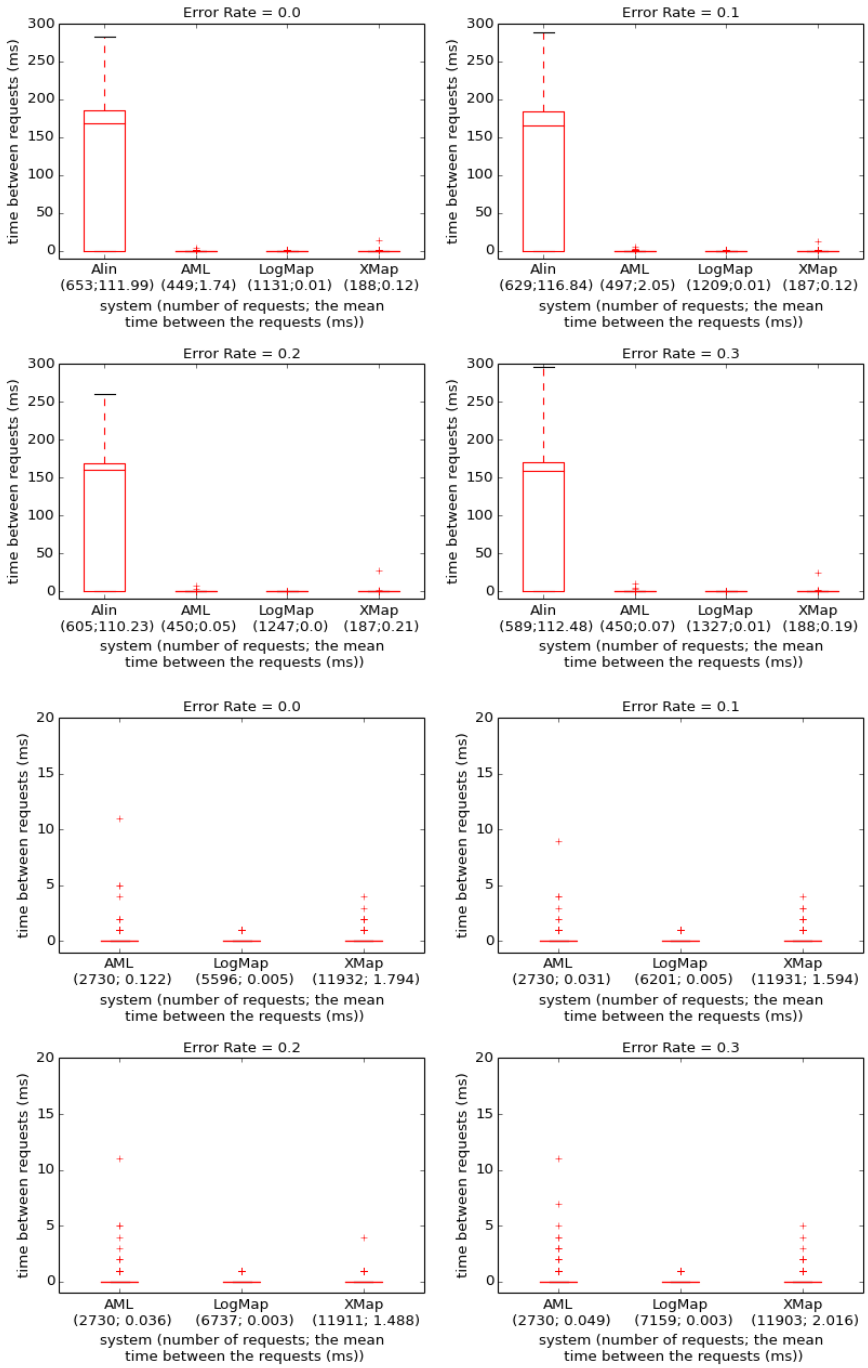


Fig. 7. Time intervals between requests to the user/oracle for the FMA-NCI (top 4 plots) and SNOMED-NCI (bottom 4 plots) datasets from the LargeBio track. Whiskers: Q1-1,5IQR, Q3+1,5IQR, IQR=Q3-Q1. The labels under the system names show the number of requests and the mean time between the requests.

produced by the systems are typically larger than without interaction, which makes the repair process harder. The introduction of oracle errors complicates the process further, and may make an alignment irreparable if the system follows the oracle's feedback blindly.

9 Instance matching

The instance matching track aims at evaluating the performance of matching tools when the goal is to detect the degree of similarity between pairs of items/instances expressed in the form of OWL Aboxes. The track is organized in two independent tasks called *SYNTHETIC* and *DOREMUS*. Each test is based on two datasets called source and target and the goal is to discover the matching pairs (i.e., mappings) among the instances in the source dataset and the instances in the target dataset.

For the sake of clarity, we split the presentation of he task results in two different subsections.

9.1 SYNTHETIC task

Task data The SYNTHETIC datasets are produced using SPIMBENCH [40] with the aim to generate descriptions of the same entity where *value-based*, *structure-based* and *semantics-aware* transformations are employed on source data in order to create the target data.

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.

We stress that an instance in the source dataset can have none or one matching counterpart in the target dataset. A dataset is composed of a Tbox and a corresponding Abox. Source and target datasets share almost the same Tbox (differences are found in the properties due to the employed structure-based transformations). The *Sandbox* scale is 10K triples $\approx$ 380 instances while the *Mainbox* scale is 50K triples $\approx$ 1800 instances. We asked the participants to match the creative works (news items, blogposts and programmes) in the source dataset against the instances of the corresponding class in the target dataset.

Results The participants of the SYNTHETIC task are the AgreementMakerLight (AML), I-Match, Legato and LogMap systems. In order to evaluate those systems we built a ground truth containing the set of expected links where an instance i_1 in the source dataset is associated with an instance j_1 in the target dataset that has been generated as a modified description of i_1. The value-based, structure-based and semantics-aware transformations were applied on different triples of the source dataset pertaining to one class instance.

The systems were judged on the basis of the *precision*, *recall* and *F-measure* results shown in Table 19. LogMap and Legato produce links that are very often correct (resulting in a good precision) but fail to capture a large number of the expected links (resulting in a lower recall). In the case of AML and I-Match systems, the probability of capturing a correct link is high, but the probability of a retrieved link to be correct is lower, resulting in a high (almost perfect) recall but a low precision. Regarding the size of the dataset, LogMap and Legato systems have better results for the Sandbox dataset. On the other hand, AML and I-Match systems exhibit the same performance for both the Sandbox and Mainbox datasets.

Table 19. SYNTHETIC task results

System	Sandbox task			Mainbox task		
	Precision	**Recall**	**F-measure**	**Precision**	**Recall**	**F-measure**
AML	0.849	1.000	0.918	0.855	1.000	0.922
I-Match	0.854	0.997	0.920	0.856	0.997	0.921
Legato	0.980	0.730	0.840	0.970	0.700	0.810
LogMap	0.938	0.763	0.841	0.893	0.709	0.790

9.2 DOREMUS task

Task data The DOREMUS task, having its second appearance at the OAEI, contains real world datasets coming from two major French cultural institutions – The BnF (French National Library) and the PP (Philharmonie de Paris). The data are about classical music works and follow the DOREMUS model (one single vocabulary for both datasets) issued from the DOREMUS project.[14] Each data entry, or instance, is a bibliographical record about a musical piece, containing properties such as the composer, the title(s) of the work, the year of creation, the key, the genre, the instruments, to name a few. These data have been converted to RDF from their original UNI- and INTER-MARC formats and anchored to the DOREMUS ontology and a set of domain controlled vocabularies by the help of the *marc2rdf* converter,[15] developed for this purpose within the DOREMUS Project (for more details on the conversion method and on the ontology we refer to [1] and [31]). Note that these data are highly heterogeneous. We have selected works described both at the BnF and at the PP with different degrees of heterogeneity in their descriptions. The datasets have been selected for the purposes of two sub-tasks.

Heterogeneities (HT): This sub-task consists in aligning two datasets, BnF-1 and PP-1, containing about 238 instances each, by discovering 1:1 equivalence relations between them. There are different types of heterogeneities that these data manifest, identified by music library experts, such as multilingualism, differences in catalogs, differences in spelling, different degrees of description, etc. The goal is to test the ability of linking tools to cope with these heterogeneities. The participants are asked to map only instances of the $F22\_Self-Contained\_Expression$ class.

False Positives Trap (FPT): This sub-task consists in correctly disambiguating the instances contained in two datasets of small sizes (75 instances each), BnF-2 and PP-2, by discovering 1:1 equivalence relations between the instances that they contain. Librarian experts have selected several groups of music works with highly similar descriptions across the two datasets, where there exist only one correct match in each group. The goal is to challenge the linking tools capacity to avoid the generation of false positives and match correctly instances in the presence of highly similar but yet distinct candidates. The participants are asked to map only instances of the $F22\_Self-Contained\_Expression$ class.

Results Five systems participated and returned results on the DOREMUS track: AML, I-Match, *Legato*, LogMap and NjuLink. Two systems stand out, outperforming significantly the other participants on both sub-tasks – Legato and NjuLink, both achieving F-measures of over 0.9

[14] `http://www.doremus.org`
[15] `https://github.com/DOREMUS-ANR/marc2rdf`

(NjuLink leading on HT and Legato - on FP-trap). Both tasks appear to be fairly challenging for the majority of the systems, with average F-measures of 0.636 for HT task and 0.565 for the FP-trap task.

Table 20. Results of the DOREMUS task

System	HT task			FP-Trap task		
	Precision	**Recall**	**F-measure**	**Precision**	**Recall**	**F-measure**
AML	0.851	0.479	0.613	0.914	0.427	0.582
I-Match	0.680	0.071	0.129	1.00	0.053	0.101
Legato	0.930	0.920	**0.930**	1.00	0.980	**0.990**
LogMap	0.406	0.882	0.556	0.119	0.880	0.210
NjuLink	0.966	0.945	**0.955**	0.959	0.933	**0.946**

10 HOBBIT Link Discovery

In this track, two benchmark generators are proposed to deal with *link discovery* for spatial data represented as *trajectories* i.e., sequences of longitude, latitude pairs. This new track is using the HOBBIT platform[16] and follows different instructions than the SEALS-based tracks.

We use TomTom[17] datasets in order to create the benchmark. TomTom datasets contain representations of traces (GPS fixes). Each trace consists of a number of points. Each point has a timestamp, longitude, latitude and speed. The points are sorted in ascending order by the timestamp of the corresponding GPS fix. Each task of the HOBBIT Link Discovery Track is composed of two datasets with different number of instances to match, namely the Sandbox and the Mainbox.

The HOBBIT Link Discovery track comprises of two tasks:

– *Task 1 (Linking)* measures how well the systems can match traces that have been modified using string-based approaches along with addition and deletion of intermediate points. Since TomTom datasets only contain coordinates, in order to apply string-based modifications implemented in LANCE [41] we have replaced a number of those points with labels retrieved from Linked Data spatial datasets using the Google Maps[18], Foursquare[19] and Nominatim Openstreetmap[20] APIs. This task also contains modifications on date and coordinate formats. An instance in the source dataset has one matching counterpart in the target dataset. For the *Linking Task*, the Sandbox scale is 100 instances while the Mainbox scale is 5K instances. We asked the participants to match traces in the source and the target datasets.
The participants of the Linking task are AgreementMakerLight (AML) and OntoIdea systems. For evaluation, we built a ground truth containing the set of expected links where an instance i_1 in the source dataset is associated with an instance j_1 in the target dataset that has been generated as an altered description of i_1.
The way that the transformations were done, was to apply value-based, and structure-based transformations on different triples pertaining to instances of class Trace.

[16] https://project-hobbit.eu/outcomes/hobbit-platform/
[17] https://www.tomtom.com/
[18] https://developers.google.com/maps/
[19] https://developer.foursquare.com/
[20] http://nominatim.openstreetmap.org/

Table 21. HOBBIT Link Discovery Linking Task

System	Precision	Recall	F-measure	Run Time
Sandbox task				
AML	1.000	1.000	1.000	11722
OntoIdea	0.990	0.990	0.990	19806
Mainbox task				
AML	1.000	1.000	1.000	134456
OntoIdea	Platform Time Limit (75 mins)			

The systems were judged on the basis of *precision*, *recall*, *F-measure* and *runtime* results that are shown in Table 21. Both AML and OntoIdea systems return high precision and recall capturing all the correct links. Regarding runtime, for the Sandbox dataset, AML needs less time than OntoIdea and for the Mainbox dataset, AML completes the task with perfect results in contrast to OntoIdea that was not able to complete it and stopped when it hit the platform time limit (75 mins). Datasets, reference alignments, and task results are available on the HOBBIT website: `https://project-hobbit.eu/challenges/om2017/`.

– *Task 2 (Spatial)* measures how well the systems can identify the DE-9IM (Dimensionally Extended nine-Intersection Model) topological relations. The supported spatial relations are the following: *Equals, Disjoint, Touches, Contains/Within, Covers/CoveredBy, Intersects, Crosses, Overlaps*. The traces 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.

Given a LineString source geometry s, a LineString target geometry t and a DE-9IM topological relation r, we ask the participants to match an instance from s with one or more instances in t such as their Intersection Matrix follows the definition of r. For evaluation, we built a ground truth using RADON [42] containing the set of expected links where an instance i_1 in the source dataset is associated with one or more instances in the target dataset that has been generated as an altered description of i_1. For the *Spatial Task*, the Sandbox scale is 10 instances and the Mainbox scale is 2K instances.

The participants to the Spatial task are AgreementMakerLight (AML), OntoIdea, Rapid Discovery of Topological Relations (RADON) and Silk systems.

The systems were judged on the basis of *precision*, *recall*, *F-measure* and *runtime* results shown in Table 22 and Figures 8 and 9. We should mention that we are only presenting the time performance and not precision, recall and f-measure as all were equal to 1.0 except *OntoIdea* that reports for the *Touches* and *Overlaps* relations value 0.99. Moreover, Silk is not participating in relations *Covers* and *Covered By* and OntoIdea is not participating in relation *Disjoint*.

From the results we can observe that:

- **OntoIdea** has the best performance in the Sandbox dataset *but* in the Mainbox dataset the runtime increases and the system seems to not be able to handle large datasets easily.
- **Silk** also seems to have a similar behaviour as **OntoIdea**.
- **RADON** and **AML** systems seem to handle the growth of the dataset size smoother.
- **AML** does not provide any results for the *Disjoint* relation since it reaches the platform time limit

Datasets, reference alignments, and task results are available on the HOBBIT website: `https://project-hobbit.eu/challenges/om2017/`.

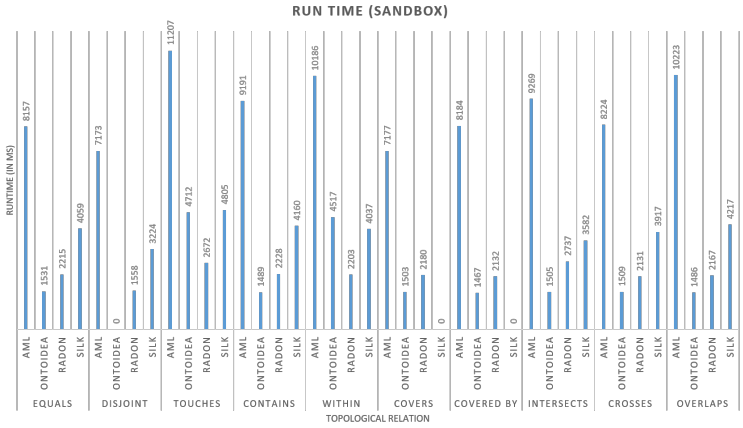


Fig. 8. HOBBIT Link Discovery Spatial Task (Sandbox)

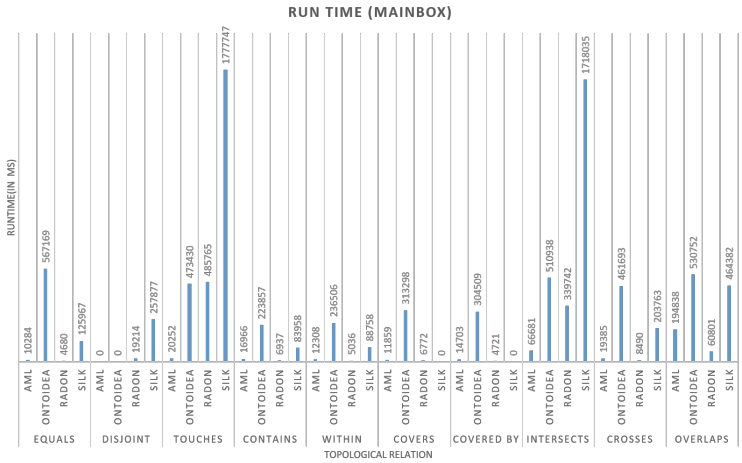


Fig. 9. HOBBIT Link Discovery Spatial Task (Mainbox)

11 Process Model Matching

In 2013 and in 2015 the community, interested in business process modeling conducted an evaluation campaign similar to the OAEI [4]. Instead of matching ontologies, the task was to match process models described in different formalisms like BPMN and Petri Nets. Within this track we offer a subset of the tasks from the Process Model Matching Contest as OAEI track by converting the process models to an ontological representation. By offering this track, we hope to gain insights in how far ontology matching systems are capable of solving the more specific problem of matching process models. This track is also motivated by the discussions at the end of the 2015 Ontology Matching workshop, where many participants showed their interest in such a track.

Table 22. Spatial Benchmark results

Relation	System	Sandbox Run Time	Mainbox Run Time
EQUALS	AML	8157	10284
	OntoIdea	1531	567169
	RADON	2215	4680
	Silk	4059	125967
DISJOINT	AML	7173	Time-out (75 min)
	OntoIdea	Not participating	
	RADON	1558	19214
	Silk	3224	257877
TOUCHES	AML	11207	20252
	OntoIdea	4712	473430
	RADON	2672	485765
	Silk	4805	1777747
CONTAINS	AML	9191	16966
	OntoIdea	1489	223857
	RADON	2228	6937
	Silk	4160	83958
WITHIN	AML	10186	12308
	OntoIdea	4517	236506
	RADON	2203	5036
	Silk	4037	88758
COVERS	AML	7177	11859
	OntoIdea	1503	313298
	RADON	2180	6772
	Silk	Not participating	
COVERED BY	AML	8184	14703
	OntoIdea	1467	304509
	RADON	2132	4721
	Silk	Not participating	
INTERSECTS	AML	9269	66681
	OntoIdea	1505	510938
	RADON	2737	339742
	Silk	3582	1718035
CROSSES	AML	8224	19385
	OntoIdea	1509	461693
	RADON	2131	8490
	Silk	3917	203763
OVERLAPS	AML	10223	194838
	OntoIdea	1486	530752
	RADON	2167	60801
	Silk	4217	464382

11.1 Experimental Settings

We used two datasets from the 2015 Process Matching Contest. The first dataset (University Admission dataset) deals with processing applications of Master students to a university. It consists

of nine different process models where each describes the concrete process of a specific German university. We already used that dataset in the 2016 edition of the OAEI. The models are encoded as BPMN process models. We converted the BPMN representation of the process models to a set of assertions (ABox) using the vocabulary defined in the BPMN 2.0 ontology (TBox). The second dataset, known as the Birth Registration dataset, describes the process of registering a new born child in different countries. The process models were originally available as Petri Nets. We converted them also to an ABox in an ontological representation. For that reason the resulting matching tasks are instance matching tasks where each ABox is described by the same TBox.

For each pair of processes manually generated reference alignments are available. Typical activities within that domain are *Sending acceptance*, *Invite student for interview*, or *Wait for response*. These examples illustrate one of the main differences to the ontology matching task. The labels are usually verb-object phrases that are sometimes extended with more words. Another important difference is related to the existence of an execution order (i.e., the model is a complex sequence of activities) which can be understood as the counterpart to a type hierarchy.

Only three systems generated non-empty results when running them against our datasets. These systems are AML, LogMap, and I-Match. Note that we tried to execute all systems marked as instance matching systems. However, the other systems threw exceptions or produced empty alignments. We have collected all generated non-empty alignments. These alignments are the raw results that the following report is based on.

In our evaluation, we computed standard precision and recall, as well as the harmonic mean known as f-measure. The dataset we used consists of several test cases. We aggregated the results and present the micro average results. The gold standard we used for our first set of evaluation experiments is based on the gold standard that has also been used at the Process Model Matching Contest in 2015 [4]. We modified only some minor mistakes (resulting in changes less than 0.5 percentage points). In order to compare the results to the results obtained by the process model matching community, we present also the recomputed values of the submissions to the 2015 contest.

We extent our evaluation ("Standard" in Tables 23 and 24) by an evaluation measure that makes use of a non-binary reference alignment ("Probabilistic" in Tables 23 and 24). This probabilistic measure is based on a gold standard which is manually and independently generated by several domain experts. The number of votes of these annotators are applied as support values in the probabilistic evaluation. For a detailed discussion, please refer to [29].

Furthermore, we evaluate the matching systems via matching patterns. Therefore the matching task as well as the matcher output is automatically categorized into categories with different complexity level. We classified each alignment in one out of five categories exclusively. In this way, strength and weaknesses of the matching systems can be analysed. For more details we refer to [30].

11.2 Results

The following tables show the results of our evaluation. Participants of the Process Model Matching Contest and the OAEI 2016 edition are depicted in gray font, while this years OAEI participants are shown in black font. Note that some systems participated with a version that has not been modified with respect to its results comparing the OAEI 2016 and 2017 submission. We added only one entry for them with the label OAEI-16/17. This is only the case for the first dataset, which we have used already in 2016.

Tables 23 and 24 summarize the results of our evaluation. "P" abbreviates precision, "R" is recall, "FM" stands for f-measure and "Rk" means rank. The prefix "Pro" indicates the probabilistic versions of the precision, recall, f-measure and the associated rank. The OAEI participants

are ranked on position 1, 11, 12 with an overall number of 17 systems listed in the table (when using the standard metrics). Note that AML-PM at the PMMC 2015 was a matching system that was based on a predecessor of AML participating at the OAEI 2016. The good results of AML are surprising, since we expected that matching systems specifically developed for the purpose of process model matching would outperform ontology matching systems applied to the special case of process model matching. While AML contains also components that are specifically designed for the process matching task (a flooding-like structural matching algorithm), its relevant main components are developed for ontology matching and the sub-problem of instance matching. AML and LogMap achieve the same results as in 2016. I-Match participates in 2017 for the first time. Compared to the results of the tools specialized for the problem of process model matching, the results of I-Match are still very good. There are still five systems that have in particular been designed for matching process models, which achieve worse results.

Table 23. Results of the Process Model Matching track for the University Admission dataset

Participant			Standard				Probabilistic			
System	**Contest**	**Size**	**P**	**R**	**FM**	**Rk**	**ProP**	**ProR**	**ProFM**	**Rk**
AML	OAEI-16/17	221	0.719	0.685	0.702	1	0.742	0.283	0.410	2
AML-PM	PMMC-15	579	0.269	0.672	0.385	15	0.377	0.398	0.387	4
BPLangMatch	PMMC-15	277	0.368	0.440	0.401	13	0.532	0.272	0.360	8
DKP	OAEI-16	177	0.621	0.474	0.538	8	0.686	0.219	0.333	9
DKP*	OAEI-16	150	0.680	0.440	0.534	9	0.772	0.211	0.331	10
KnoMa-Proc	PMMC-15	326	0.337	0.474	0.394	14	0.506	0.302	0.378	5
KMatch-SSS	PMMC-15	261	0.513	0.578	0.544	6	0.563	0.274	0.368	7
LogMap	OAEI-16/17	267	0.449	0.517	0.481	11	0.594	0.291	0.390	3
I-Match	OAEI-17	192	0.521	0.431	0.472	12	0.523	0.183	0.271	16
Match-SSS	PMMC-15	140	0.807	0.487	0.608	4	0.761	0.192	0.307	12
OPBOT	PMMC-15	234	0.603	0.608	0.605	5	0.648	0.258	0.369	6
pPalm-DS	PMMC-15	828	0.162	0.578	0.253	17	0.210	0.335	0.258	17
RMM-NHCM	PMMC-15	220	0.691	0.655	0.673	2	0.783	0.297	0.431	1
RMM-NLM	PMMC-15	164	0.768	0.543	0.636	3	0.681	0.197	0.306	13
RMM-SMSL	PMMC-15	262	0.511	0.578	0.543	7	0.516	0.242	0.329	11
RMM-VM2	PMMC-15	505	0.216	0.470	0.296	16	0.309	0.294	0.301	14
TripleS	PMMC-15	230	0.487	0.483	0.485	10	0.486	0.210	0.293	15

The results for the Birth Registration dataset are more interesting, because we are using this dataset in 2017 for the first time. Moreover, the dataset contains a higher amount of correspondences that are hard to find by comparing the labels on a lexical level. This results usually in a significantly lower F-measure compared to the University Admission dataset.

The results show that AML is no longer the best of all matching systems. Four systems from the process matching community achieve better results in terms of f-measure. This dataset is dominated by the OPBOT system, while AML is among a group of follow-up systems that perform still significantly better than the rest of the field. The other two systems, LogMap and I-Match, achieve close results which are slightly worse than the average results. It is interesting to see that the ranking among the three systems is the same across the two datasets.

In the probabilistic evaluation, in the University Admission dataset however, the OAEI participants gain position 2, 3, 16 respectively. LogMap rises from position 11 to 3. The (probabilistic) precision improves over-proportionally for this matcher, because LogMap generates many corre-

Table 24. Results of the Process Model Matching track for the Birth Registration dataset

Participant			Standard				Probabilistic			
System	**Contest**	**Size**	**P**	**R**	**FM**	**Rk**	**ProP**	**ProR**	**ProFM**	**Rk**
AML	OAEI-17	502	0.454	0.391	0.420	5	0.467	0.515	0.490	10
AML-PM	PMMC-15	503	0.423	0.365	0.392	7	0.513	0.505	0.509	7
BPLangMatch	PMMC-15	279	0.645	0.309	0.418	6	0.661	0.417	0.511	5
KnoMa-Proc	PMMC-15	740	0.234	0.297	0.262	15	0.224	0.437	0.296	15
KMatch-SSS	PMMC-15	185	0.800	0.254	0.385	8	0.865	0.379	0.527	4
LogMap	OAEI-17	239	0.615	0.252	0.358	11	0.834	0.411	0.551	3
I-Match	OAEI-17	188	0.734	0.237	0.358	12	0.812	0.366	0.504	8
Match-SSS	PMMC-15	128	0.922	0.202	0.332	13	0.974	0.315	0.476	11
OPBOT	PMMC-15	383	0.713	0.468	0.565	1	0.650	0.517	0.576	1
pPalm-DS	PMMC-15	490	0.502	0.422	0.459	2	0.469	0.521	0.493	9
RMM-NHCM	PMMC-15	267	0.727	0.333	0.456	3	0.781	0.443	0.565	2
RMM-NLM	PMMC-15	128	0.859	0.189	0.309	14	0.912	0.293	0.443	14
RMM-SMSL	PMMC-15	354	0.508	0.309	0.384	9	0.518	0.42	0.464	13
RMM-VM2	PMMC-15	492	0.474	0.400	0.433	4	0.454	0.48	0.466	12
TripleS	PMMC-15	266	0.613	0.280	0.384	10	0.651	0.426	0.515	6

spondences which are not included in the binary gold standard but are included in the probabilistic one. The ranking of LogMap demonstrates that a strength of the probabilistic metric lies in the broadened definition of the gold standard where weak mappings are included but softened (via the support values). In the probabilistic evaluation for the Birth Registration dataset, the three participating matchers gain ranking 3, 8 and 10. LogMap rises from rank 11 to 3 in the probabilistic evaluation. The matcher LogMap mainly identifies correspondences with high support (of which many are not included in the binary gold standard). For the matcher AML, the opposite effect can be observed. The matcher AML does not profit as much from the broadened gold standard in the probabilistic evaluation in the Birth Registration dataset compared to the other matching systems. The matchers improve their performance compared to the binary evaluation. This indicates that in the binary gold standard many reasonable alignments are missing. Thus the matchers improve their performance with the probabilistic evaluation. For details about the probabilistic metric, please refer to [29].

The results indicate that the progress made in ontology matching has also a positive impact on other related matching problems, like it is the case for process model matching. While it might require to reconfigure, adapt, and extend some parts of the ontology matching systems, such a system seems to offer a good starting point which can be turned with a reasonable amount of work into a good process matching tool. We have to emphasize that only three participants decided to apply their systems to the new track of process model matching. Thus, we have to be cautious to generalize the results we observed so far.

To allow for an in-depth analysis of the performance of the matching systems, we make use of a new evaluation method which automatically classifies the matching task into matching patterns with different attributes. The matching patterns are assigned automatically to the reference alignment, as well as to the matcher output of the three participating matchers. Then category-dependent precision, recall and f-measure are computed for each category separately. For more details please refer to [30].

Tables 25 and 26 show the results of the matching systems for each of the categories. The second column, the f-measure (FM) over all matching patterns, is given as the micro value, i.e. it

Approach	FM	Cat. trivial [44.3%][103]			Cat. I no word iden. [29.3%][68]			Cat. II one verb iden. [11.6%][27]			Cat. III one word iden. [7.3%][17]			Cat. misc [7.3%][17]		
		cP	cR	cFM	cP	cR	cFM	cP	cR	cFM	cP	cR	cFM	cP	cR	cFM
AML	.702	.890	.942	.915	.953	.603	.739	.833	.185	.303	.667	.353	.462	.167	.529	.254
I-Match	.472	.907	.942	.924	–	–	–	.400	.074	.125	–	–	–	.500	.059	.105
LogMap	.481	.894	.981	.935	–	–	–	.500	.148	.229	.133	.353	.194	.089	.529	.153

Table 25. Results assigned to matching patterns of University Admission dataset

Approach	FM	Cat. trivial [4.5%][26]			Cat. I no word iden. [74.9%][437]			Cat. II one verb iden. [1.5%][9]			Cat. III one word iden. [9.9%][58]			Cat. misc [9.1%][53]		
		cP	cR	cFM	cP	cR	cFM	cP	cR	cFM	cP	cR	cFM	cP	cR	cFM
AML	.420	.759	.846	.800	.427	.364	.393	.133	.222	.167	.438	.362	.396	.632	.453	.527
I-Match	.358	.950	.731	.826	.746	.236	.358	.667	.222	.333	.400	.103	.164	.667	.151	.246
LogMap	.358	.339	.731	.463	.726	.261	.384	–	–	–	.357	.086	.139	.818	.170	.281

Table 26. Results assigned to matching patterns of Birth Registration dataset

is computed over all test cases. The remaining columns provide the category-dependent precision (cP), recall (cR) and f-measure (cFM) for each matcher in each category. cP, cR and cFM are macro values, independently computed for each category. Moreover, for each category, the tables contain in the heading the fraction of correspondences from the whole data set as well as the total number of correspondences of a category in the reference alignment. Cat. I contains alignments which have no word in common (syntactically). It can be observed that for the University Admission dataset it is sufficient to identify mainly trivial correspondences. I-Match and LogMap do not compute any alignments of the most complex category ("Cat. I"). However, AML has a very high performance for "Cat. I". In the Birth Registration dataset the fraction of trivial alignments is very low. The most dominant category is "Cat. I". Therefore, it is not sufficient to focus on the identification of trivial alignments. In contrast to the University Admission datatset, the matchers compute reasonable alignments from "Cat. I" in the Birth Registration dataset. The low performance of the three matchers for "Cat. trivial" in the Birth Registration dataset indicates mistakes in the binary gold standard.

11.3 Conclusions

In 2016 we organized the Process Model Matching track for the first time. Our evaluation effort was motivated by the idea that Ontology Matching methods and techniques can also be used in the related field of Process Model Matching. For that reason we converted one (and in 2017 two) of the most prominent Process Model matching test datasets into an ontological representation. The resulting matching problems are instance matching tasks.

While we were aware that an instance matching system will not be able to exploit the sequential aspects of the given process models out of the box, we expected lexical components to generate results that are already on an acceptable level. Even though some of the systems generated very good results, overall only a few of the systems participating at the OAEI were capable of generating any results for our test cases. We still do not fully understand the reasons for this outcome.

In order to facilitate the evaluation process for participants which cannot evaluate their matchers with SEALS, we developed a web-based evaluation platform[21] to potentially increase the number of participants. This platform was intended to be used by potential participants from the process matching community that are not interested in an OAEI participation, which is tailored for ontology matching systems. Within this platform, participants are able to select one or multiple gold standards for one of the datasets and subsequently upload their corresponding matcher results. Afterwards, the participants are able to select from a variety of different metrics including not only different types of precision, recall and f-measure but also general statistics for the generated output. Unfortunately, no further matching systems participated via the platform.

The participation rate indicates that only a limited number of participants is interested in process model matching. For that reason we will not offer a third edition of this track in 2018.

12 Statistical analysis

The traditional evaluation carried out in the OAEI tracks consists simply of comparing and ranking systems based on performance scores such as F-measure. In the case of tracks with multiple datasets, performance scores are averaged for all datasets, and the systems are compared accordingly. While performance scores enable us to gage the performance of matching systems individually, they are insufficient for drawing statistically meaningful comparisons between systems.

In the interest of providing a more in-depth comparison of the matching systems that participated in this year's competition, this section presents an analysis based on statistical inference.

12.1 Methods

For one-dataset comparisons, we use McNemar's test. This test takes as input the alignments produced by two matching systems plus the reference alignment, and produces as output an indicator which shows if either system is better than the other or whether they are approximately the same. This method of comparison does not need a particular performance score to be determined beforehand. Further, the comparison is not solely based on the juxtaposition of two scalars, but rather, it is substantiated by the statistical evidence (null hypothesis testing). Two variants of McNemar's test were considered: one where false correspondences were ignored so that the comparison was predicated only on the correct correspondences found by matching systems; and another where both correct and false correspondences were considered, meaning that systems were compared based on the full alignment they generated. A directed graph can be used to visualize the outcome of the test. Interested readers are referred to [34] for more details about the utilization of this methodology.

For comparisons over multiple datasets, we used the Friedman test with the corresponding post-hoc procedure for comparison. This test requires the specification of one performance score. The outcome of the test can be visualized by critical difference (CD) diagrams.

Since the comparisons between matching systems are done pairwise, it is necessary to correct the statistics for multiple testing. We used the Bergmann correction method to control the family-wise error rate in all tests.

[21] http://alkmaar.informatik.uni-mannheim.de/pmmc

12.2 Results

Anatomy track In this year's competition, 11 systems participate in the anatomy track. However, the alignments of the LogMap family could not be parsed by the Alignment API, so we had to leave them out from the comparative analysis for this track.

Figure 10 shows the directed graph with the outcome of McNemar's test over participatory systems when the false correspondences are not taken into account. Figure 11 shows the corresponding result when all correspondences are considered. The nodes in these graphs are the systems and a directed edge $A \rightarrow B$ indicates the superiority of A over B. If there is no such an edge between any two systems, then they are claimed to be more or less equivalent.

According to these figures, AML is the best system and Wiki3 and ALIN are the bottom ones, from both perspectives. There are two differences between the two approaches to conducting the test. SANOM outperforms KEPLER when the false correspondences are not considered, and KEPLER is better than SANOM if wrong correspondences are taken into account. It means that SANOM discovers more correct correspondences than KEPLER, but also more false correspondences. A similar pattern holds for the comparison of POMAP and YAM-BIO. Interestingly, no systems are declared to be equivalent, so the outcome of McNemar's test is similar to a ranking scheme.

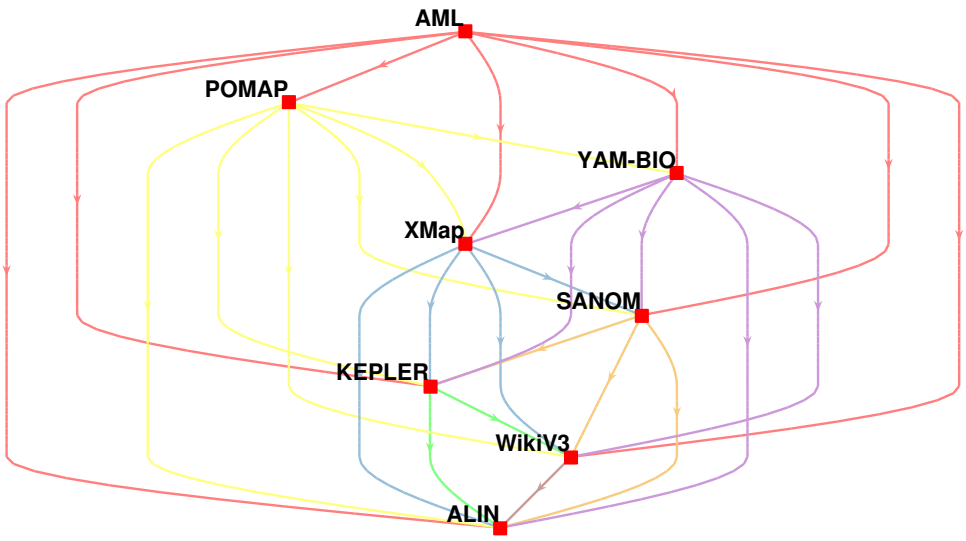


Fig. 10. Comparison of alignment systems participated in OAEI 2017 on the anatomy track while the false correspondences are not considered.

Conference track This track consists of 21 small matching tasks between 7 different ontologies. Three different types of matching are considered: (i) M1: only matching the classes; (i) M2: only matching the properties; (ii) M3: matching both classes and properties. The reference alignment has also three different variants. Hence, there are nine different modes of evaluating systems, based on the type of matching and the type of reference alignment. The Friedman test

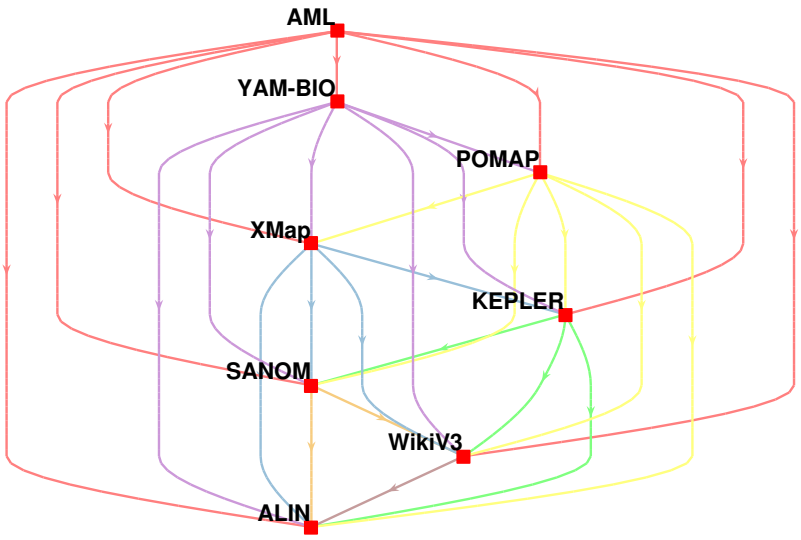


Fig. 11. Comparison of alignment systems participated in OAEI 2017 on the anatomy track while the false correspondences are taken into account.

was applied considering the F-measure of the systems on each of the 21 tasks for each of the evaluation modes.

Figure 12 shows the CD diagram of the systems that participated in this track. In this figure, the x axis is the average rank obtained by the Friedman test, and the systems with the same performance are connected to each other by the red lines. The lower the average rank in the CD plot, the better the performance of the system.

The CD diagram for this track provides little information and insight about the difference between systems, likely due to the small sample size for the comparison (systems produce only between 90 and 240 correspondences in total in this track). What is readily seen from this plot is the superiority of AML, LogMap, and XMap and the poor performance of ALIN, SANOM, and POMap.

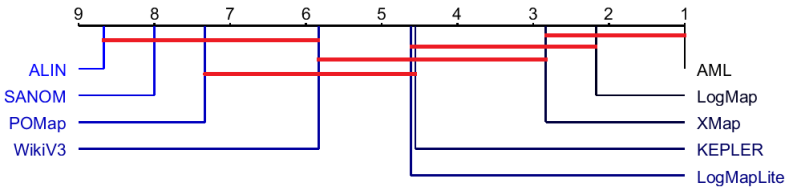


Fig. 12. Comparison of alignment systems participated in OAEI 2017 on the Conference track. The x-axis is the average rank of each system obtained by the Friedman test. Systems which are not significantly different from each other are connected by the red lines.

LargeBio track This track consists six matching tasks of large size. The Friedman test was applied to the F-measure obtained by each system over each alignment task. Figure 13 shows the corresponding CD diagram for this track. According to this plot, the group containing AML, XMap, YAM-BIO, LogMap, and LogMapBio are the best systems, and POMAP, SANOM, and KEPLER are the systems with lackluster performance in this track.

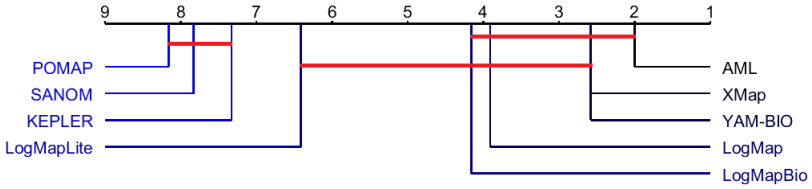


Fig. 13. Comparison of alignment systems participated in OAEI 2017 on the LargeBio track. The x-axis is the average rank of each system obtained by the Friedman test. Systems which are not significantly different from each other are connected by the red lines.

Multifarm track This track involves 55 matching tasks with ontologies from different languages. The Friedman test was applied to the F-measure obtained by each system over each task. The CD diagram depicting the outcome of the test is shown in Figure 14.

According to this graph, AML is exclusively the best alignment system in this track. LogMap, CroLOM, and KEPLER perform equally better than the remaining systems. At the other extreme, LogMapLite, XMap, and SANOM show a poor performance in this track, while WikiV3 ranks in between the two trios.

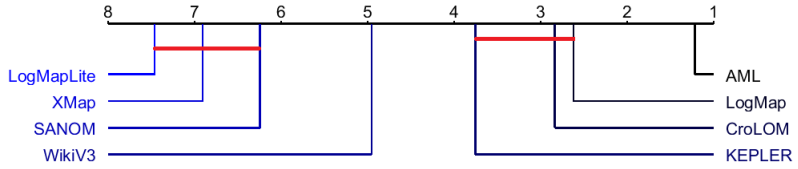


Fig. 14. Comparison of alignment systems participated in OAEI 2017 on the Conference track. The x-axis is the average rank of each system obtained by the Friedman test. Systems which are not significantly different from each other are connected by the red lines.

13 Lesson learned and suggestions

The lessons learned from running OAEI 2017 were the following:

A) Like last year, this year we requested tool registration in June and preliminary submission of wrapped systems by the end of July, but were more strict in its enforcement. As a result, we recorded the smallest number of errors and incompatibilities with the SEALS client during the evaluation phase in recent OAEI editions.

B) As has been the trend, some system developers struggled to get their systems working with the SEALS client, mostly due to incompatible versions of libraries. While participation on the new HOBBIT track was relatively low due to the novelty of the HOBBIT platform and the short deadline for systems to adapt to it, the solution of using Docker containers to wrap systems seems promising, and we are already looking into phasing out the SEALS client in favour of the HOBBIT platform.

C) While the number of participants this year was similar to that of recent years, their distribution through the tracks was uneven. The expressive ontologies tracks had no shortage of participants, and still a fair number participated in the more specialized multifarm track. However, participation in the interactive matching track and in the three instance matching tracks (process model, instance, and hobbit) was underwhelming. The latter is puzzling considering the prize sponsored by IBM Research for the system with the best performance across the instance matching tracks. Granted, the division of instance matching tracks between the SEALS client and the HOBBIT platform did not help their cause, as of the 7 total systems that participated in instance matching tasks, only 2 made both a SEALS and a HOBBIT submission. Nevertheless, the division between "traditional" ontology matching and instance matching is readily apparent, as only 2 systems have participated in both track families.

D) In previous years we identified the need for considering non-binary forms of evaluation, namely in cases where there is uncertainty about some of the reference mappings. A first non-binary evaluation type was implemented in the Conference track in 2015, followed by Disease and Phenotype, and Process Model in 2016. This year, we have introduced statistical tests to compare matching systems, an analysis that was carried out on the results of 4 tracks. This approach provides more insights into the comparative performance of systems as well as more statistical rigour, and thus we hope that it can be expanded and fully integrated into the OAEI tracks in future editions.

The lessons learned in the various OAEI 2017 track were the following:

conference: Since there have been no improvement in matchers performance this year from the perspective of performed evaluation modalities we will consider to add or replace existing evaluation modalities for future editions of OAEI to help disclose further matchers characteristics.

largebio: While the current reference alignments, with incoherence-causing mappings flagged as uncertain, make the evaluation fair to all systems, they are only a compromise solution, not an ideal one. Thus, we should aim for manually repairing and validating the reference alignments for future editions.

phenotype: This track attracted a similar level of participation this year compared to last, despite no cash prize, which demonstrates its intrinsic value and interest among the community of ontology matching algorithm developers.

interactive: This track's participation has remained low, as most systems participating in OAEI opt to focus exclusively on fully automatic matching. We hope to draw more participants to this track in the future and will continue to expand it so as to better approximate real user interactions.

process model: The results of the Process Model track have shown that the participating ontology matching systems are capable of generating good results for the specific problem of process model matching, even though few were able to exploit the sequential aspects of the process models. Even though we offered an alternative evaluation process for participants which cannot evaluate their matchers with SEALS, this alternative failed to attract further participants. The low participation rate in this track indicates that only a limited number of

participants is interested in process model matching. For that reason we will not offer a third edition of this track in 2018.

instance: In order to attract more instance matching systems to participate in value semantics (val-sem), value structure (val-struct), and value structure semantics (val-struct-sem) tasks, we need to produce benchmarks that have fewer instances (in the order of 10000), of the same type (in our benchmark we asked systems to compare instances of different types). To balance those aspects, we must then produce benchmarks that contain more complex transformations.

14 Conclusions

The OAEI 2017 saw the same number of participants as in recent years, with a healthy mix of new and returning systems. While last year we posited that new participants were drawn by the allure of prize money in the new Disease and Phenotype track, the evidence this year seems to contradict it. On the one hand, participation in Disease and Phenotype remain high this year despite no prize money. On the other hand, the prize money on offer for performance in instance matching did not attract many participants to those tracks. Nevertheless, the fact that there continues to be corporate interest in ontology matching to the point of offering prize money bodes well for the future of the OAEI.

Like last year, judging from the repeated tracks, there has been no substantial progress to the state of the art in ontology matching overall this year:

- There was no noticeable improvement with regard to system run times.
- There were few improvements with regard to F-measure, with the top results in most tracks remaining the same.
- There was no significant progress with regard to the ability of matching systems to handle large ontologies and datasets, either in traditional ontology matching or in instance matching.
- There was no progress with regard to alignment repair systems, with only a few returning systems employing them.

This conclusion may be due to a plateau being reached by matching systems in some tracks, and investing in improving results further would bring diminishing returns. However, it is also the case that long-term participants tend to focus more on the new datasets and tracks on offer than on improving in repeated tracks. Given the variety of tracks on offer, it is difficult for system developers to aim at improving across all tracks each year.

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

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 [43]. 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 would also like to thank IBM Research for sponsoring the instance matching tracks by offering prize money for the best performing systems.

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 data set.

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

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

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

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

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

Ernesto Jimenez-Ruiz has also been partially supported by the BIGMED project (IKT 259055), the HealthInsight project (IKT 247784), the SIRIUS Centre for Scalable Data Access (Research Council of Norway, project no.: 237889).

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.

References

1. Manel Achichi, Rodolphe Bailly, Cécile Cecconi, Marie Destandau, Konstantin Todorov, and Raphaël Troncy. Doremus: Doing reusable musical data. In *ISWC PD: International Semantic Web Conference Posters and Demos*, 2015.
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 *Proc. 11th ISWC ontology matching workshop (OM), Kobe (JP)*, 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 *Proc. 7th ISWC ontology matching workshop (OM), Boston (MA, US)*, pages 73–115, 2012.

4. Gonçalo Antunes, Marzieh Bakhshandeh, José Borbinha, João Cardoso, Sharam Dadashnia, Chiara Di Francescomarino, Mauro Dragoni, Peter Fettke, Avigdor Gal, Chiara Ghidini, Philip Hake, Abderrahmane Khiat, Christopher Klinkmüller, Elena Kuss, Henrik Leopold, Peter Loos, Christian Meilicke, Tim Niesen, Catia Pesquita, Timo Péus, Andreas Schoknecht, Eitam Sheetrit, Andreas Sonntag, Heiner Stuckenschmidt, Tom Thaler, Ingo Weber, and Matthias Weidlich. The process model matching contest 2015. In *6th EMISA Workshop*, pages 127–155, 2015.
5. Benhamin Ashpole, Marc Ehrig, Jérôme Euzenat, and Heiner Stuckenschmidt, editors. *Proc. K-Cap Workshop on Integrating Ontologies*, Banff (Canada), 2005.
6. Olivier Bodenreider. The unified medical language system (UMLS): integrating biomedical terminology. *Nucleic Acids Research*, 32:267–270, 2004.
7. Caterina Caracciolo, Jérôme Euzenat, Laura Hollink, Ryutaro Ichise, Antoine Isaac, Véronique Malaisé, Christian Meilicke, Juan Pane, Pavel Shvaiko, Heiner Stuckenschmidt, Ondrej Sváb-Zamazal, and Vojtech Svátek. Results of the ontology alignment evaluation initiative 2008. In *Proc. 3rd ISWC ontology matching workshop (OM), Karlsruhe (DE)*, pages 73–120, 2008.
8. Michelle Cheatham, Zlatan Dragisic, Jérôme Euzenat, Daniel Faria, Alfio Ferrara, Giorgos Flouris, Irini Fundulaki, Roger Granada, Valentina Ivanova, Ernesto Jiménez-Ruiz, et al. Results of the ontology alignment evaluation initiative 2015. In *Proc. 10th ISWC ontology matching workshop (OM), Bethlehem (PA, US)*, pages 60–115, 2015.
9. Bernardo Cuenca Grau, Zlatan Dragisic, Kai Eckert, Jérôme Euzenat, Alfio Ferrara, Roger Granada, Valentina Ivanova, Ernesto Jiménez-Ruiz, Andreas Oskar Kempf, Patrick Lambrix, Andriy Nikolov, Heiko Paulheim, Dominique Ritze, François Scharffe, Pavel Shvaiko, Cássia Trojahn dos Santos, and Ondrej Zamazal. Results of the ontology alignment evaluation initiative 2013. In Pavel Shvaiko, Jérôme Euzenat, Kavitha Srinivas, Ming Mao, and Ernesto Jiménez-Ruiz, editors, *Proc. 8th ISWC ontology matching workshop (OM), Sydney (NSW, AU)*, pages 61–100, 2013.
10. 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.
11. Jérôme David, Jérôme Euzenat, François Scharffe, and Cássia Trojahn dos Santos. The alignment API 4.0. *Semantic web journal*, 2(1):3–10, 2011.
12. Cássia Trojahn dos Santos, Bo Fu, Ondrej Zamazal, and Dominique Ritze. State-of-the-art in multilingual and cross-lingual ontology matching. In *Towards the Multilingual Semantic Web, Principles, Methods and Applications*, pages 119–135. 2014.
13. Zlatan Dragisic, Kai Eckert, Jérôme Euzenat, Daniel Faria, Alfio Ferrara, Roger Granada, Valentina Ivanova, Ernesto Jiménez-Ruiz, Andreas Oskar Kempf, Patrick Lambrix, Stefano Montanelli, Heiko Paulheim, Dominique Ritze, Pavel Shvaiko, Alessandro Solimando, Cássia Trojahn dos Santos, Ondrej Zamazal, and Bernardo Cuenca Grau. Results of the ontology alignment evaluation initiative 2014. In *Proc. 9th ISWC ontology matching workshop (OM), Riva del Garda (IT)*, pages 61–104, 2014.
14. Zlatan Dragisic, Valentina Ivanova, Patrick Lambrix, Daniel Faria, Ernesto Jiménez-Ruiz, and Catia Pesquita. User validation in ontology alignment. In *The Semantic Web - ISWC 2016 - 15th International Semantic Web Conference, Kobe, Japan, October 17-21, 2016, Proceedings, Part I*, pages 200–217, 2016.
15. Zlatan Dragisic, Valentina Ivanova, Huanyu Li, and Patrick Lambrix. Experiences from the anatomy track in the ontology alignment evaluation initiative. *Journal of Biomedical Semantics*, 2017.
16. Jérôme Euzenat, Alfio Ferrara, Laura Hollink, Antoine Isaac, Cliff Joslyn, Véronique Malaisé, Christian Meilicke, Andriy Nikolov, Juan Pane, Marta Sabou, François Scharffe, Pavel Shvaiko, Vassilis Spiliopoulos, Heiner Stuckenschmidt, Ondrej Sváb-Zamazal, Vojtech Svátek, Cássia Trojahn dos Santos, George Vouros, and Shenghui Wang. Results of

the ontology alignment evaluation initiative 2009. In *Proc. 4th ISWC ontology matching workshop (OM), Chantilly (VA, US)*, pages 73–126, 2009.
17. Jérôme Euzenat, Alfio Ferrara, Christian Meilicke, Andriy Nikolov, Juan Pane, François Scharffe, Pavel Shvaiko, Heiner Stuckenschmidt, Ondrej Sváb-Zamazal, Vojtech Svátek, and Cássia Trojahn dos Santos. Results of the ontology alignment evaluation initiative 2010. In *Proc. 5th ISWC ontology matching workshop (OM), Shanghai (CN)*, pages 85–117, 2010.
18. Jérôme Euzenat, Alfio Ferrara, Robert Willem van Hague, Laura Hollink, Christian Meilicke, Andriy Nikolov, François Scharffe, Pavel Shvaiko, Heiner Stuckenschmidt, Ondrej Sváb-Zamazal, and Cássia Trojahn dos Santos. Results of the ontology alignment evaluation initiative 2011. In *Proc. 6th ISWC ontology matching workshop (OM), Bonn (DE)*, pages 85–110, 2011.
19. Jérôme Euzenat, Antoine Isaac, Christian Meilicke, Pavel Shvaiko, Heiner Stuckenschmidt, Ondrej Svab, Vojtech Svatek, Willem Robert van Hage, and Mikalai Yatskevich. Results of the ontology alignment evaluation initiative 2007. In *Proc. 2nd ISWC ontology matching workshop (OM), Busan (KR)*, pages 96–132, 2007.
20. 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.
21. Jérôme Euzenat, Malgorzata Mochol, Pavel Shvaiko, Heiner Stuckenschmidt, Ondrej Svab, Vojtech Svatek, Willem Robert van Hage, and Mikalai Yatskevich. Results of the ontology alignment evaluation initiative 2006. In *Proc. 1st ISWC ontology matching workshop (OM), Athens (GA, US)*, pages 73–95, 2006.
22. Jérôme Euzenat and Pavel Shvaiko. *Ontology matching*. Springer-Verlag, Heidelberg (DE), 2nd edition, 2013.
23. Daniel Faria, Ernesto Jiménez-Ruiz, Catia Pesquita, Emanuel Santos, and Francisco M. Couto. Towards Annotating Potential Incoherences in BioPortal Mappings. In *13th International Semantic Web Conference*, volume 8797 of *Lecture Notes in Computer Science*, pages 17–32. Springer, 2014.
24. 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*, 2018.
25. Ernesto Jiménez-Ruiz and Bernardo Cuenca Grau. LogMap: Logic-based and scalable ontology matching. In *Proc. 10th International Semantic Web Conference (ISWC), Bonn (DE)*, pages 273–288, 2011.
26. 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.
27. Ernesto Jiménez-Ruiz, Christian Meilicke, Bernardo Cuenca Grau, and Ian Horrocks. Evaluating mapping repair systems with large biomedical ontologies. In *Proc. 26th Description Logics Workshop*, 2013.
28. Yevgeny Kazakov, Markus Krötzsch, and Frantisek Simancik. Concurrent classification of EL ontologies. In *Proc. 10th International Semantic Web Conference (ISWC), Bonn (DE)*, pages 305–320, 2011.
29. Elena Kuss, Henrik Leopold, Han Van der Aa, Heiner Stuckenschmidt, and Hajo A Reijers. Probabilistic evaluation of process model matching techniques. In *Conceptual Modeling: 35th International Conference, ER 2016, Gifu, Japan, November 14-17, 2016, Proceedings 35*, pages 279–292. Springer, 2016.
30. Elena Kuss and Heiner Stuckenschmidt. Automatic classification to matching patterns for process model matching evaluation. In *Proceedings of the ER Forum 2017 and the ER 2017 Demo Track co-located with the 36th International Conference on Conceptual Modelling (ER 2017), Valencia, Spain, - November 6-9, 2017.*, pages 292–305, 2017.

31. Pasquale Lisena, Manel Achichi, Eva Fernández, Konstantin Todorov, and Raphaël Troncy. Exploring linked classical music catalogs with overture. In *ISWC PD: International Semantic Web Conference Posters and Demos*, 2016.
32. Christian Meilicke. *Alignment Incoherence in Ontology Matching*. PhD thesis, University Mannheim, 2011.
33. 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.
34. Majid Mohammadi, Amir Ahooye Atashin, Wout Hofman, and Yaohua Tan. Comparison of ontology alignment algorithms across single matching task via the McNemar test. *arXiv*, arXiv:1704.00045.
35. Boris Motik, Rob Shearer, and Ian Horrocks. Hypertableau reasoning for description logics. *Journal of Artificial Intelligence Research*, 36:165–228, 2009.
36. Heiko Paulheim, Sven Hertling, and Dominique Ritze. Towards evaluating interactive ontology matching tools. In *Proc. 10th Extended Semantic Web Conference (ESWC), Montpellier (FR)*, pages 31–45, 2013.
37. Catia Pesquita, Daniel Faria, Emanuel Santos, and Francisco Couto. To repair or not to repair: reconciling correctness and coherence in ontology reference alignments. In *Proc. 8th ISWC ontology matching workshop (OM), Sydney (AU)*, page this volume, 2013.
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 Couto. Ontology alignment repair through modularization and confidence-based heuristics. *CoRR*, abs/1307.5322, 2013.
40. T. Saveta, E. Daskalaki, G. Flouris, I. Fundulaki, M. Herschel, and A.-C. Ngonga Ngomo. Pushing the limits of instance matching systems: A semantics-aware benchmark for linked data. In *WWW, Companion Volume*, 2015.
41. Tzanina Saveta, Evangelia Daskalaki, Giorgos Flouris, Irini Fundulaki, Melanie Herschel, and Axel-Cyrille Ngonga Ngomo. Lance: Piercing to the heart of instance matching tools. In *International Semantic Web Conference*, pages 375–391. Springer, 2015.
42. Mohamed Ahmed Sherif, Kevin Dreßler, Panayiotis Smeros, and Axel-Cyrille Ngonga Ngomo. RADON - Rapid Discovery of Topological Relations. In *Proceedings of The Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)*, 2017.
43. 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.
44. Alessandro Solimando, Ernesto Jiménez-Ruiz, and Giovanna Guerrini. Detecting and correcting conservativity principle violations in ontology-to-ontology mappings. In *The Semantic Web–ISWC 2014*, pages 1–16. Springer, 2014.
45. Alessandro Solimando, Ernesto Jimenez-Ruiz, and Giovanna Guerrini. Minimizing conservativity violations in ontology alignments: Algorithms and evaluation. *Knowledge and Information Systems*, 2016.
46. York Sure, Oscar Corcho, Jérôme Euzenat, and Todd Hughes, editors. *Proc. ISWC Workshop on Evaluation of Ontology-based Tools (EON), Hiroshima (JP)*, 2004.

Montpellier, Dayton, Linköping, Grenoble, Lisboa,
Milano, Heraklion, Kent, Oslo, Mannheim, Amsterdam,
Delft, Trento, Toulouse, Prague
December 2017

ALIN Results for OAEI 2017

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

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

Abstract. ALIN is an ontology alignment system specialized in the interactive alignment of ontologies. Its main characteristic is the selection of correspondences to be shown to the expert, depending on the previous feedbacks given by the expert. This selection is based on semantic and structural characteristics. 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 2017 competition and discusses its results.

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

1 Presentation of the system

A large amount of data repositories became available due to the advances in information and communication technologies. Those repositories, however, are highly semantically heterogeneous, which hinders their integration. Ontology alignment has been successfully applied to solve this problem, by discovering correspondences between two distinct ontologies which, in turn, conceptually define the data stored in each repository. Among the various ontology alignment approaches that exist in the literature, interactive ontology alignment includes the participation of experts of the domain to improve the quality of the final alignment. This approach has proven more effective than non-interactive ontology alignment [1]. ALIN is an ontology alignment system specialized in interactive alignment.

1.1 State, purpose, general statement

ALIN is an ontology alignment system, specialized in the ontology interactive alignment, based primarily on linguistic matching techniques, using the Wordnet as external resource. After generating an initial set of correspondences (called set of candidate correspondences, which are the correspondences selected to receive the feedback from the expert), interactions are made with the expert, and to each interaction, the set of candidate correspondences is modified. The modification of the set of candidate correspondences is through the use of the structural analysis of ontologies and use of correspondence anti-patterns. The interactions continue until there are no more candidate correspondences left. ALIN was built with a special focus on the interactive matching track of OAEI 2017.

1.2 Specific techniques used

The ALIN algorithm is shown in algorithm 1.

Algorithm 1 ALIN algorithm

Input: Two ontologies to be aligned
Output: Alignment between the two ontologies
1: Loading of ontologies
2: Generation of the initial set of candidate correspondences
3: Automatic classification of correspondences
4: Removal of correspondences by the low value of semantic similarity
5: **while** Set of candidate correspondences is not empty **do**
6: Choose correspondences to show to the expert
7: Receive expert feedback to chosen correspondences and remove them of the set of candidate correspondences
8: Remove correspondences in an correspondence anti-pattern from set of candidate correspondences
9: Insert some data property and object property correspondences into set of candidate correspondences
10: Insert some correspondences from the backup set into set of candidate correspondences
11: **end while**

The steps of ALIN algorithm are the following:

1. Load of the ontologies with load of classes, object properties and data properties through the Align API[1]. For each entity some data are stored such as name and label. In the case of classes, their superclasses and disjunctions are saved. In the case of object properties the properties that are their hypernyms and their associated classes are saved. The classes of data properties are saved, too. ALIN does not use instances. The ALIN can only work with ontologies whose entity names are in English.

2. As an initial set of candidate correspondences a stable marriage algorithm with incomplete preference lists with maximum size of the list equals to 1, using linguistic metrics to sort the priority list was used [2]. The list is sorted in decreasing order. For this algorithm only the correspondence whose first entity is in the list of second entity and vice-versa is selected. The linguist metrics used are Jaccard, Jaro-Winkler and n-Gram [3] provided by Simmetrics API[2] and

[1] Alignment API . Available at http://alignapi.gforge.inria.fr/ Last accessed on Oct, 10, 2017.
[2] String Similarity Metrics for Information Integration . Available on http://www.coli.uni-saarland.de/courses/LT1/2011/slides/stringmetrics.pdf. Last accessed on Oct, 10, 2017.

Resnick, Jiang-Conrath and Lin [3] provide by HESML API[3] that use Wordnet. To use Wordnet the canonical form of the entity names is needed, therefore Stanford CoreNLP API[4] was used. The most frequent synsets of words are used to calculate semantic similarities. To find this synset is used the WS4J API[5]. The algorithm is run six times, once by each metric, and the result set is the union of results of each metric.

3. The value of the similarity metrics (Resnick, Jiang-Conrath, Lin, Jaccard, Jaro-Winkler and n-Gram) vary from 0 to 1 (1 is the maximum value). When a correspondence in the set of candidate correspondences has all the six metrics with the maximum value, it is added to the final alignment and removed from the set of candidate correspondences. There are exceptions to this rule, some correspondences that fall into some structural patterns are not put on the final alignment and are not removed from the set of candidate correspondences.

4. The correspondences whose entities has one of its linguistic metrics less than a given threshold are removed from the set of candidate correspondences. These correspondences are put into a backup set, and can return to the set of candidate correspondences using structural analysis. The use of this technique can best be seen in [4], with the difference that, in [4], instead of applying a threshold, it was removed the classes of correspondences that were not in the same Wordnet synset.

5-11. At this point the interactions with the expert begin. The correspondences in the set of candidate correspondences are sorted by the sum of similarity metric values, with the greatest sum first. The correspondences are showed to the expert. The set of candidate correspondences has, at first, only correspondences of classes. When the expert answer one question, the set of candidate correspondences is modified. Correspondences (besides the correspondence answered by expert) can be removed and correspondences can be included into the set of candidate correspondences, depending on the answer of the expert. If the expert does not accept the correspondence it is removed from the set of candidate correspondences. But if the expert accepts the correspondence it is removed from the set of candidate correspondences and put in the final alignment.

At each interaction with the expert:

- We remove from the set of candidate correspondences and disregard all the correspondences that are in correspondence anti-pattern [5] with the correspondences accepted by the expert;

- We insert into the set of candidate correspondences, data property and object property correspondences related to the class correspondences accepted by the expert.

[3] HESML. Available at https://www.researchgate.net/publication/313881253_HESML_A_scalable_ontologybased_semantic_similarity_measures_library_with_a_set_of_reproducible_experiments_and_a_replication_dataset Last accessed on Oct, 10, 2017.

[4] Stanford CoreNLP . Available at http://stanfordnlp.github.io/CoreNLP/ Last accessed on Oct, 10, 2017.

[5] WS4J . Available at https://github.com/Sciss/ws4j Last accessed on Nov, 08, 2017.

- We insert into the set of candidate correspondences, correspondences of the backup set (step 4) whose both entities are subclasses of the classes of a correspondence accepted by expert.

This step continues until the set of candidate correspondences is empty.

Detailed information about the ALIN system can be seen in the master thesis of Jomar da Silva[6].

1.3 Link to the system and parameters file

ALIN is available through Google drive (

https://drive.google.com/open?id=1myVtcRoKKdUDHQTKNKsomna8AFbukanf) as a package for running through the SEALS client.

2 Results

The system ALIN has been developed with its focus on interactive ontology alignment. The approach performs better when the number of data and object properties is proportionately large. ALIN considers properties associated to correspondent classes when selecting entities for user feedback, thus allowing for increased recall. When the number of properties in the ontologies is small, the system still generates a very precise alignment, but its recall tends to decrease.

Another characteristic of ALIN is its reliance on an interactive phase. The non-interactive phase of the system is quite simple, mainly based on maximum string similarity, specializing in maintaining a high precision without worrying about recall, generating initially a low f-measure. The recall increases in the interactive phase. Finally, ALIN is also not robust to users errors. The system uses a number of techniques that take advantage of the expert feedback to reach other conclusions. When the expert gives a wrong answer it is propagated generating other errors, thereby decreasing the f-measure.

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

As expected the participation of ALIN in non-interactive alignment processes showed the following results: high precision and not so high recall, as can be seen in Anatomy track[7] shown in Table 1, where recall+ field refers to non-trivial correspondences found and Coherent field filled by + indicates that the generated alignment is consistent.

[6] INTERACTIVE ONTOLOGY ALIGNMENT: AN APPROACH BASED ON THE INTERACTIVE MODIFICATION OF THE SET OF CANDIDATE CORRESPONDENCES . Available at http://www2.uniriotec.br/ppgi/banco-de-dissertacoes-ppgi-unirio/ano-2017/interactive-ontology-alignment-an-approach-based-on-the-interactive-modification-of-the-set-of-candidate-correspondences/view Last accessed on Nov, 12, 2017.

[7] Results for OAEI 2017 - Anatomy track . Available at http://oaei.ontologymatching.org/2017/results/anatomy/index.html Last accessed on Nov, 012, 2017.

Regarding the Conference track[8], as ALIN evaluates only the properties associated with classes already evaluated as belonging to the alignment, the alignment of the M2 type (which take into account only the properties of ontologies) were with the f-measure = 0, as can be seen in Table 2. As properties are evaluated only in the interactive phase in the ALIN, alignments of type M1 (only classes) remained with a higher recall than M3 (classes and properties), as can be seen in Table 2, because the reference alignments of type M3 contain properties besides classes.

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

Runtime	Size	Precision	F-Measure	Recall	Recall+	Coherent
836	516	0.996	0.506	0.339	0.0	+

Table 2. Participation of ALIN in Conference non-interactive track

	Threshold	Precision	Recall	F1-Measure	F2-Measure	F.5-Measure
ra1+m1	0.0	0.89	0.32	0.47	0.37	0.66
ra1+m2	0.0	0.0	0.0	0.0	0.0	0.0
ra1+m3	0.0	0.89	0.27	0.41	0.31	0.61

2.2 Comments on the participation of the ALIN in interactive tracks

Table 3. Participation of ALIN in Anatomy interactive track - Error rate 0.0

Tool	Run Time (sec)	Precision	Recall	F-measure	Total Requests	Distinct Mappings
ALIN	1074	0.993	0.794	0.882	939	1472
AML	45	0.968	0.948	0.958	241	240
LogMap	23	0.982	0.846	0.909	388	1164
XMap	43	0.927	0.865	0.895	35	35

[8] "Results of Evaluation for the Conference track within OAEI 2017 . Available at http://oaei.ontologymatching.org/2017/conference/eval.html Last accessed on Nov, 12, 2017.

Table 4. Participation of ALIN in Anatomy interactive track - Error rate 0.1

Tool	Run Time (sec)	Precision	Recall	F-measure	Total Requests	Distinct Mappings
ALIN	1000	0.94	0.745	0.831	905	1352
AML	45	0.956	0.946	0.95	266	264
LogMap	23	0.962	0.83	0.891	388	1164
XMap	44	0.927	0.865	0.895	35	35

Table 5. Participation of ALIN in Conference interactive track - Error rate 0.0

Tool	Run Time (sec)	Precision	Recall	F-measure	Total Requests	Distinct Mappings
ALIN	35	0.957	0.731	0.829	329	571
AML	30	0.912	0.711	0.799	271	270
LogMap	35	0.886	0.61	0.723	82	246
XMap	21	0.837	0.57	0.678	4	4

Table 6. Participation of ALIN in Conference interactive track - Error rate 0.1

Tool	Run Time (sec)	Precision	Recall	F-measure	Total Requests	Distinct Mappings
ALIN	35	0.804	0.669	0.73	321	549
AML	30	0.841	0.701	0.765	282	275
LogMap	35	0.851	0.598	0.702	82	246
XMap	21	0.837	0.57	0.678	4	4

Anatomy track In this track the program ALIN showed the highest precision among the four evaluated tools when the error rate is zero, as can be seen in Table 3. When the error rate increases both the precision as the recall falls, reducing the f-measure, as can be seen in Table 4. This is expected and explained earlier.

As ontologies of the Anatomy Track contains almost no properties, some interactive techniques used in ALIN can not be utilized, like the selection of properties associated with classes with positive feedback. This has limited the increase in recall, which influenced the f-measure.

Conference Track In this track ALIN stood out, showing the greatest f-measure among the four tools when the error rate is zero, as can be seen in 5, as with a loss of f-measure when the error rate increases, as can be seen in Table 6.

Other results, including results with other error rates can be seen on the OAEI 2017[9] page.

[9] Results for OAEI 2017 - Interactive Track . Available at http://oaei.ontologymatching.org/2017/results/interactive/index.html Last accessed on Nov, 11, 2017.

2.3 Comparison of the participation to ALIN in OAEI 2017 with his participation in OAEI 2016

The difference between the participation of ALIN in OAEI 2016 and his participation in OAEI 2017 was the use of the HESML API in 2017 instead of the WS4J API in calculating semantic similarities, which greatly increased the efficiency in these calculations. In ALIN's participation in OAEI 2016[6], three semantic similarity metrics were used: Wu-Palmer, Jiang-Conrath and Lin. In ALIN's participation in OAEI 2017 the metrics Resnick, Jiang-Conrath and Lin were used. Resnick's exchange of Wu-Palmer is due to the fact that the Wu-Palmer metric in the HESML API took longer to execute than the same metric in the WS4J API. The Resnick metric proved to be much faster than the Wu-Palmer metric in the HESML API and according to [7] as good as, so the Resnick metric was chosen to take Wu-Palmer's place in the implementation of ALIN at OAEI 2017. More information about the HESML API can be found in [8]. In table 7. it can be seen that the ALIN runtime has decreased considerably with the use of the HESML API instead of the WS4J API. In the Anatomy interactive track of OAEI 2016, ALIN did not use the semantic metrics, only the string metrics, since the semantic metrics were taking a long time, making it impossible to execute it. In OAEI 2017, using the HESML API, it was possible to use semantic metrics, which led to an increase in the quality of the alignment generated, but with an increase in the expert's participation. The execution time also increased with the inclusion of semantic metrics, as we can see in table 8.

Table 7. Participation of ALIN in Conference interactive track - OAEI 2016/2017- Error rate 0.0

Year	Run Time (sec)	Precision	Recall	F-measure	Total Requests	Distinct Mappings
2016	101	0.957	0.735	0.831	326	574
2017	35	0.957	0.731	0.829	329	571

Table 8. Participation of ALIN in Anatomy interactive track - OAEI 2016/2017- Error rate 0.0

Year	Run Time (sec)	Precision	Recall	F-measure	Total Requests	Distinct Mappings
2016	505	0.993	0.749	0.854	803	1221
2017	1074	0.993	0.794	0.882	939	1472

3 General Comments

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

(a) handling user error rate;

(b) generating a higher quality (especially w.r.t. recall) initial alignment in its non-interactive phase;

(c) reducing the number of interactions with the expert; and

(d) optimize the process to reduce its execution time, especially in alignments with large numbers of correspondences, such as Anatomy.

3.1 Conclusions

Within certain characteristics, the ALIN system stands out in ontology alignment process in interactive application scenarios, especially when the amount of data and object properties are relatively large and when the expert does not make mistakes. With these features there is an alignment generated with relatively high precision and recall.

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

References

1. H. Paulheim, S. Hertling, and D. Ritze, Towards Evaluating Interactive Ontology Matching Tools, Lect. Notes Comput. Sci., vol. 7882, pp. 31-45, 2013.
2. R. W. Irving, D. F. Manlove, and G. OMalley, Stable marriage with ties and bounded length preference lists J. Discret. Algorithms, vol. 7, no. 2, pp. 213-219, 2009.
3. J. Euzenat and P. Shvaiko, Ontology Matching - Second Edition, 2. Springer-Verlag, 2013.
4. Silva, J., Baião, F. A., Revoredo, K., & Euzenat, J. (n.d.). Semantic Interactive Ontology Matching : Synergistic Combination of Techniques to Improve the Set of Candidate Correspondences.
5. A. Guedes, F. Baião, e K. Revoredo, Digging Ontology Correspondence Antipatterns, Proceeding WOP14 Proc. 5th Int. Conf. Ontol. Semant. Web Patterns, vol. 1302, p. 3848, 2014.
6. J. Silva, F. A. Baião, and K. Revoredo, ALIN Results for OAEI 2016, CEUR Workshop Proc., vol. 1766, 2016.
7. E. G. M. Petrakis, G. Varelas, A. Hliaoutakis, and P. Raftopoulou, Design and Evaluation of Semantic Similarity Measures for Concepts Stemming from the Same or Different Ontologies object instrumentality, Proc. 4th Work. Multimed. Semant., vol. 4, pp. 233-237, 2006.
8. Lastra-Díaz, J. J., García-Serrano, A., Batet, M., Fernández, M., & Chirigati, F. (2017). HESML: A scalable ontology-based semantic similarity measures library with a set of reproducible experiments and a replication dataset. Information Systems, 66, 97118. http://doi.org/10.1016/j.is.2017.02.002

Results of AML in OAEI 2017

Daniel Faria[1], Booma Sowkarthiga Balasubramani[2], Vivek R. Shivaprabhu[2], Isabela Mott[3], Catia Pesquita[3], Francisco M. Couto[3], and Isabel F. Cruz[2]

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

Abstract. AgreementMakerLight (AML) is an automated ontology matching system that was developed with both extensibility and efficiency in mind. This paper describes its configuration for the OAEI 2017 competition and discusses its results. For this OAEI edition, we built upon the instance matching foundations we laid last year, and tackled the new Hobbit track and its new evaluation platform. AML was the only system to participate in all OAEI tracks this year, and was the top performing system or among the top performing ones in nearly all tracks, including the new Hobbit track. It was awarded the IBM Research prize for the best performing system in all instance matching related tracks.

1 Presentation of the System

1.1 State, Purpose, General Statement

AgreementMakerLight (AML) is an ontology matching system inspired by AgreementMaker [2, 3] but more concerned with efficiency, in order to tackle large-scale matching problems [7]. While it originally focused primarily on the biomedical domain, it has since been expanded to address a broad range of ontology and instance matching problems. AML relies heavily on lexical matching techniques [10], with an emphasis on the use of background knowledge [6], but also includes structural components for both matching and filtering—namely it features a logical repair algorithm [11].
This year, our development of AML centered on the instance matching tasks from the new Hobbit track, and to a lesser degree on the new tasks in the Process Model Matching and Instance Matching tracks.
We maintained the solution of using configuration files we adopted last year, but only for the instance matching tasks, as only for these is the goal of the matching tasks not always inferable from the datasets (e.g., it is generally not possible to infer when the goal is to match only instances of a given type).

1.2 Specific Techniques Used

For the sake of brevity, this section focuses mainly on the features of AML that are new for this edition of the OAEI. For a complete description of AML's matching strategy, please refer to the results papers of the last two OAEI editions [4, 5].

1.2.1 AML-Hobbit

The Hobbit track datasets required profound adaptations to AML. First, although the ontology files were included in the training sets, in the Hobbit client only the instances were provided to the matching systems. This meant that the datasets could not be correctly parsed using OWL API [8], and required us to create an N-Triples parser tailored to these datasets (i.e., with the contextual information from the ontology files hard-coded into the parser). Second, the unusual characteristics of the matching tasks, which involve matching traces based on their geographical points, required that we implement dedicated data structures and matching algorithms.

Linking

The Linking task focused on finding equivalent traces by matching their geographical points. The information available for points could include geographical coordinates, address, timestamp, and velocity. The target dataset resulted from a transformation of the source dataset, where some information was omitted and other were altered. Of particular note was the conversion of the geographical coordinates to different coordinate systems. This required us to do the reverse conversion to the decimal system, which we performed during parsing.
The main difficulty of the task was its size, as each trace included on average ≈ 2000 points, and the full task consisted in matching 10000 traces. An efficient matching strategy was therefore paramount.
To enable such a strategy, we adopted a *HashMap*-based data structure with inverted indexes, analogous to AML's other matching structures, but where geographical points were used as keys. To this end, we defined a hash code for points based on the combination of their coordinates. This made it possible to find matching points in *O(1)* time and therefore match the trace datasets in *O(n)* time, with *n* being the total number of points in the ontology with the least points. We used the address and timestamp of the points to filter the matches, and found the velocity to be unnecessary.

Spatial

The Spatial tasks focused on determining whether traces were related according to a number of different topological relations (e.g., contains, crosses, disjoint). In this case, the traces were given as a list of coordinate pairs corresponding to their points, and no transformation of the data was necessary.
To tackle these tasks, we adopted the ESRI Geometry API, which can be used for constructing geometries and performing spatial operations and topological relationship tests on them.

1.2.2 AML-SEALS

Only a few changes were made to AML's matching strategy for the SEALS tracks since the OAEI 2016 edition [5].

Ontology Parser

We made a few changes to AML's ontology parser to cope with typical omissions in instance matching datasets, such as undeclared properties. By default, the OWL API

interprets undeclared properties to be annotation properties, which leads to erroneous parsing of the dataset, and hinders AML's performance.
Additionally, we also modified the ontology parser to process OBO logical definitions directly from OWL, as the new versions of the Disease and Phenotype track datasets already included these definitions (last year they did not, and that required us to use external files with the definitions).

Translator

We improved AML's Translator by adding a translation to English of the input ontologies in addition to the reciprocal translation we were already performing. This not only increases the likelihood that a direct match can be found between ontology entities, but also enables the use of WordNet [9].

1.3 Adaptations made for the evaluation

The Hobbit submission of AML is, as a whole, an adaptation made for the evaluation, as the specificities of the Hobbit evaluation (namely the absence of a Tbox) and the tasks (which are almost exclusively based on spatial coordinates) demanded a dedicated submission.
In addition, as in previous years, our SEALS submission included precomputed 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 in the Anatomy track was the same as last year, with 95% precision, 93.6% recall, 94.4% F-measure, and 83.2% recall+. It remains the best performing system in this track.

2.2 Conference

AML's performance in the Conference track was also the same as last year. It remains the best performing system in this track, with the highest F-measure on the full reference alignment 1 (74%), the full reference alignment 2 (70%), and on both evaluation modalities with the uncertain reference alignment (Discrete: 78%; Continuous: 77%).
Concerning the logical reasoning evaluation, AML had no consistency principle violations, but did have conservativity principle violations as this is an aspect AML deliberately doesn't take into account given that many of these violations were empirically found to be false positives.

2.3 Disease and Phenotype

AML generated 2029 mappings in the HP-MP task, 75 of which were unique. It had the highest F-measure according to the 2-vote silver standard, with 87.2%. In the HP-MeSH task, it generated 5638 mappings of which 678 were unique. It also had the highest F-measure according to the 2-vote silver standard, with 87.1%. In the HP-OMIM task, it generated 6681 mappings of which 679 were unique, and was third in F-measure with 87.8%. In the DOID-ORDO task, it generated the most mappings (4779) and the most unique mappings (1520), and as a result had a relatively low F-measure according to the 2-vote silver standard (66.1%).

2.4 Hobbit

AML produced a perfect result (100% F-measure) in Linking and all Spatial tasks, with the sole exception of the Spatial disjoint mainbox task, where it timed out. In Linking, it had the lowest run time in both the sandbox and mainbox modalities (the other participant timed out in the mainbox task). In Spatial, it had generally the highest run time in the sandbox modalities, but had the lowest run time in the mainbox modality of several tasks, which suggests that it is more scalable than the other participants.

2.5 Instance Matching

In the SPIMBENCH sub-track, AML obtained the second highest F-measure in the sandbox modality (91.8%) and the highest F-measure in the mainbox modality (92.2%). In the Doremus sub-track, AML's results were underwhelming, with only 61.3% F-measure in the Heterogeneities task and 58.2% F-measure in the False Positives Trap task. These tasks were considerably more difficult than the homonym tasks of last year.

2.6 Interactive Matching

AML had an equivalent performance to last year, as we were unable to devote time to address the issues we detected on its user interaction module. In the Anatomy dataset, AML had the highest F-measure (95.8% with 0% errors), the second lowest number of oracle requests, and the lowest impact of errors, with a drop in performance under 3% between 0 and 30% errors. In the Conference dataset, it was second in F-measure with 0% errors, but first when errors were introduced (for all error rates). Despite this, it was more impacted by errors than LogMap, due to the fact that it made considerably more user interactions.

2.7 Large Biomedical Ontologies

AML had the same results as last year in this track, except that the alignment it produced for the SNOMED-NCI whole ontologies tasks had more unsatisfiabilities. This is a consequence of the fact that this year we opted to switch off the use of the ELK

reasoner when parsing the ontologies, due to the SPIMBENCH ontologies being inconsistent. Although AML's ontology parser captures most of the subclass and equivalence relationships identified by ELK (which is why there are only differences in this task), it doesn't capture all of them. AML obtained either the highest or the second highest F-measure in all tasks, and had the highest average F-measure overall with 82.7% (ignoring the XMAP results, since this system uses the UMLS metathesaurus as background knowledge, which is the basis of the reference alignments).

2.8 Multifarm

AML improved its results in matching different ontologies, and remains the system with the highest F-measure (46%). However, its performance in matching the same ontologies decreased, and it has only the fourth best F-measure (26%). This decrease was reportedly due to some errors in parsing the alignments for which a confidence higher than 1 was generated, an issue which we will investigate and address.

2.9 Process Model

AML obtained the same result as last year in the University Admission dataset, with 70.2% F-measure. This remains the highest F-measure of all OAEI and PMMC [1] participants. In the new Birth Registration dataset, it obtained the highest F-measure among OAEI participants (42.0%), but would rank only fifth among PMMC participants.

3 General comments

3.1 Comments on the results

AML was the only system to participate in all tracks this year, and was either the best performing or among the top performing systems in nearly all tasks, including the new Hobbit track and the new datasets in the Process Model Matching and Disease and Phenotype tracks. AML was also consistently among the fastest systems and among those that produced the most coherent alignments. As was the case last year, these results reflect our continued effort to extend and improve AML while ensuring that it remains both effective and efficient.

3.2 Comments on the OAEI test cases

While we welcome the efforts of the OAEI organizers to expand it with new datasets, we must comment on some of the issues we encountered during this year's competition, and suggest some possible improvements for future editions.
In the new Hobbit track, even if it is understandable in a new massive venture such as the Hobbit evaluation platform, the tardiness of the information on the submission process and evaluation datasets hindered participation. More importantly, the fact that Tbox data was unavailable through the platform meant that participating systems had to be trained specifically to interpret the Hobbit Abox data, which we feel violates the

spirit of the OAEI.
We were also not fully satisfied with the evaluation of the Disease and Phenotype track. Generating silver standards from the alignments produced by the participating systems via voting is a reasonable starting point for producing a reference alignment, but they should not be used as-is for evaluating matching systems, as the evaluation will be unreliable and superficial. We hope that future efforts focus on improving the evaluation prior to adding more datasets.

4 Conclusion

In 2017, AML was the only system to participate in all tracks, and was among the best performing systems in nearly all tasks (with the sole exception of the Instance Matching DOREMUS sub-track). However, our efforts to participate in the new Hobbit track left little time for making other improvements to AML, and as a result, its performance in most tracks remained the same as last year. That said, our efforts were fully rewarded, as AML was awarded the IBM Research prize for the best performing system in all instance matching related tracks.

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). CP was also funded by FCT (PTDC/EEI-ESS/4633/2014). The research of IFC, BSB and VRS was partially funded by NSF awards CNS-1646395, III-1618126, CCF-1331800, and III-1213013, and by a Bill & Melinda Gates Foundation Grand Challenges Explorations grant.

References

1. G. Antunes, M. Bakhshandeh, J. Borbinha, J. Cardoso, S. Dadashnia, C. Francescomarino, M. Dragoni, P. Fettke, A. Gal, C. Ghidini, et al. The process model matching contest 2015. In *6th EMISA Workshop*, pages 127–155, 2015.
2. 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.
3. 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.
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. M. Horridge and S. Bechhofer. The owl api: A java api for owl ontologies. *Semantic Web*, 2(1):11–21, 2011.
9. G. A. Miller. WordNet: A Lexical Database for English. *Communications of the ACM*, 38(11):39–41, 1995.
10. C. Pesquita, D. Faria, C. Stroe, E. Santos, I. F. Cruz, and F. M. Couto. What's in a "nym"? Synonyms in Biomedical Ontology Matching. In *International Semantic Web Conference (ISWC)*, pages 526–541, 2013.
11. E. Santos, D. Faria, C. Pesquita, and F. M. Couto. Ontology alignment repair through modularization and confidence-based heuristics. *PLoS ONE*, 10(12):e0144807, 2015.

CroLOM results for OAEI 2017

Summary of Cross-Lingual Ontology matching Systems at OAEI

Abderrahmane Khiat

Human-Centered Computing Lab, Freie Universität Berlin, Germany
abderrahmane.khiat@fu-berlin.de, abderrahmane_khiat@yahoo.com

Abstract. This paper presents the results obtained in the OAEI 2017 campaign by our ontology matching system CroLOM. CroLOM is an automatic system especially designed for aligning multilingual ontologies. This is our second participation with CroLOM in the OAEI and the results have so far been positive.

Keywords: Cross lingual Alignment, Multilingual Ontologies Survey, Ontology Matching, Yandex, Semantic Similarity, OAEI, Direct matching.

1 Introduction

With the growing number of ontologies defined in different languages, multilingualism has become an issue of major interest in ontology matching field. Multilingual ontology alignment, defined as the process of identification of semantic correspondences between entities of different ontologies described in different natural language, represents the solution to the problem of semantic interoperability between different sources of distributed information [1, 2]. Several methods have been elaborated to semantically align multilingual ontologies. These methods can be generally split into two main categories direct and indirect matching approaches [3]. The approaches of the first category are based on external resources (i.e. translation) to align cross-lingual ontologies. However, the approaches of the second category are based on the composition of alignments such as the work proposed in [4] where the authors reuse the mappings between ontologies that already exist.

In this study, we consider the approaches of the first category, since we develop an approach which implements a direct strategy. However, there are many questions regarding this approach to address the multilingualism issue. These questions are as follows: (1) Which machine translation should be used, (2) which translation path should be considered and (3) which ontologies features and dictionaries can be exploited. In the following paragraphs, we describe the points mentioned above.

First, several translators have been developed to translate automatically the text from one natural language to another. We can mention for example: Google, Bing, SDL and Gengo translators. Each translator has its specific characteristics such as: number of source/target languages and execution time. However, selecting one or several translators (by combining them) remains an open problem. This choice is crucial in ”direct approaches”, since they apply a monolingual matching techniques in cross-lingual ontology mapping.

Second, the translation path also plays an important role to resolve the heterogeneity problem. Two translation paths can be considered, *(i)* either considering the translation path from one to another or *(ii)* selecting a pivot language which is often the English language. This choice highly depends on available sources (dictionaries, thesaurus, etc.) in different natural languages. Most matching systems consider the translation path using English as a pivot language due to available sources in English language.

Finally, in some cases, the results of a translation machine could be poor, however, to avoid this situation some ontology features can be exploited such Description Logics.

Most matching systems which implement a direct translation approach uses a well-known translators mentioned above. The current work uses also a direct matching approach. However, unlike existing approaches, it addresses the multilingualism challenge by using *(a)* the Yandex translator[1], *(b)* a translation into a pivot language after applying NLP and *(c)* a similarity computation based on the categories of the words and synonyms.

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

2 Related Work

In this section, we continue our previous work [5, 20] by covering the main cross-lingual ontology matching systems that have participated in the last editions of the Multifarm track of OAEI evaluation campaign, we should note that the Multifarm track includes the Arabic dataset [5, 6] since 2015. Most of systems which participated at OAEI use a direct translation-based matching approach.

Table 1 summarizes the results of the systems achieving the best results in the Multifarm track in previous edition. Note that some of these systems participated in different editions and they obtained low results due to problems such as parsing or accessing to translator server. These results also includes the changes that have been made on Multifarm track. The purpose of these selection is to observe the best results obtained on Multifarm track.

The AUTOMSv2 system [14] uses a free Java API named WebTranslator[2] in order to solve the multi-language problem by translating label and properties in English language. The GOMMA system [15] uses a free translation API named "mymemory"[3] to automatically translate non-English terms. The WeSeE-Match system [16] translates the fragments, labels, and comments in English as a pivot language using the Bing[4] Search APIs translation capabilities. The WikiMatch system [17] employs the

[1] https://translate.yandex.com/?lang=es-en&text=administrar&ncrnd=5317

[2] http://webtranslator.sourceforge.net/

[3] http://mymemory.translated.net/

[4] https://www.microsoft.com/en-us/translator/translatorapi.aspx

Table 1: Top systems in the multifarm track

OAEI	Top Systems	Multifarm Track	Precision	F-measure	Recall
2012	AUTOMSv2	without Arabic	0.49	0.36	0.10
2012	WeSeE	//	0.61	0.41	0.32
2012	GOMMA	//	0.29	0.31	0.36
2012	WikiMatch	//	0.34	0.27	0.23
2013	YAM++	//	0.51	0.40	0.36
2014	AML	//	0.57	0.54	0.53
2014	LogMap	//	0.80	0.40	0.28
2014	XMap	//	0.31	0.35	0.43
2015	AML		0.53	0.51	0.50
2015	LogMap		0.75	0.41	0.29
2015	XMap		0.23	0.25	0.28
2015	CLONA		0.46	0.39	0.35
2016	CroLOM		0.55	0.36	0.28
2017	KEPLER		0.43	0.31	0.25
2017	Wikiv3		0.30	0.25	0.21

Google Translation API[5] for addressing multi-lingual ontologies. The CLONA system [18] translates the entities described in different natural languages into English as a pivot language using Bing translator. Then it uses Lucene search engine and WordNet to determine alignment candidates. The XMap system [7] uses an automatic translation for obtaining correct matching pairs in multilingual ontology matching. The translation is done by querying Microsoft Translator for the full name. The AML system [8] uses an automatic translation module based on Microsoft Translator. The translation is done by querying Microsoft Translator for the full name (rather than word-by-word). To improve performance, AML stores locally all translation results in dictionary files, and queries the Translator only when no stored translation is found. The LogMap system that participated in the OAEI 2014 campaign used a multilingual module based on Google translate; however the new version of the LogMap system uses both Microsoft and Google translator APIs [11]. The YAM++ system [9] uses a multilingual translator based on Microsoft Bing to translate the annotations to English. The KEPLER and Wikiv3 systems participated for the first time at OAEI2017....

We have also observed that, at OAEI2017 the best results are still those obtained by AML system in 2015, achieving an F-measure equals to 0.51. This is surprising, in spite of many research works that have been established in the field of multilingual ontology matching.

[5] `http://code.google.com/apis/language/translate/overview.html`

3 CroLOM: Cross-Lingual Ontology Matching System

We summarize the process of our approach to provide a general idea of the proposed solution. It consists in the following successive phases:

3.1 Extraction and Normalization

CroLOM extracts first the entities of the input ontologies. Then, it employs NLP to normalize the entities described in different natural languages. Unlike existing approaches, we have applied lemmatization, stemming and stopword elimination for each natural language separately before translation step. First, for each language considered by multifarm, we have established the stop words of each language in order to eliminate them from entities labels. Second, we have developed morphological algorithms to obtain lemmatization of the entities words.

This step is important [6], since one of matchers used is (1) based on string comparison algorithm to compute similarity and (2) the categories of the words are stoked in lemma form.

3.2 Translation

Once the entities are normalized, CroLOM uses the Yandex translator in order to translate the entities described in different natural languages in English as a pivot language. After translation, CroLOM employs for the second time the normalization step in order to eliminate the stop words of the English language from entities labels.

We have mentioned before that the translation path and the used translator play important role to resolve the multilingualism heterogeneity problem. Our choice for the Yandex translator is justified by the fact that it is ranked as the 4th largest search engine in the world and it has not previously used to align multilingual ontologies. However, we have chosen English as a pivot language because there a lot dictionaries that are available in English language. These dictionaries could be exploited in order to improve our system in the future. In addition, to compute the similarity between entities, we have used dictionaries (word categories and WordNet) in English. Due to automatic translation, we have observed that some stop words can be appeared in translated entities. For this purpose, we have employed the normalization for the second time.

3.3 Similarity Computation

Once the translation and standardization are carried out, CroLOM applies first, a case conversion by converting all entities words in lower case then it passes to the similarity computation step. Unlike existing systems, which use well known matchers, we have developed a matcher which calculates the similarity between entities based on the categories of the Words, string-based algorithm and synonyms using Wordnet[7].

[6] This step allows to obtain good results such as the results of our previous work [19] (STRIM system) in instance matching.

[7] `http://wordnet.princeton.edu/`

The matcher developed establishes a Cartesian product between the two entities words, then it returns the maximum similarity value using Levenshtein distance, similarity based on WordNet and similarity based on the categories of the words. The similarity based on the categories of the words has been adapted with some modification from the project "Calculate Semantic Similarity" [8]. The project has been developed to match sentences, however we have modified the code in order to compute similarity between words.

3.4 Alignment Identification

Finally, CroLOM applies a filter to select candidate correspondences which possess the maximum similarity value in each line of Cartesian product between entities. Then it applies a second a filter to identify the correspondences that possess similarity value upper than a given threshold.

4 Experimental Study

The results obtained by running our CroLOM system on multifarm track of OAEI2017 are the same as in OAEI2016 since we partipated with the same version. The results are available at the following website: `http://oaei.ontologymatching.org/2017/results/multifarm/index.html`.

5 Conclusion

In this paper, we have presented our CroLOM system, a cross-lingual ontology matching system. CroLOM unlike existing approaches, applies first NLP on each natural language before translation. Then, it uses the Yandex translator in order to translate all entities in English as pivot language. Finally, CroLOM computes the similarity between translated entities based on the category of the words and WordNet, hybridizing statistic and semantic similarity.

As future challenges, we aim to (1) improving the quality results of our system and especially the execution time, (2) conduct a survey study that addresses all the issues mentioned above, (3) taking into account the indirect approaches.

References

1. A. Khiat and M. Benaissa, "A New Instance-Based Approach for Ontology Alignment". International Journal on Semantic Web and Information Systems (IJSWIS), Vol. 11, No. 3, ISSN 1683-3198, 2015.
2. A. Khiat and M. Benaissa, "Boosting Reasoning-Based Approach by Structural Metrics for Ontology Alignment". The Journal of Information Processing Systems (JIPS), 2015.

[8] `https://sourceforge.net/projects/semantics/`

3. S Zhang and O. Bodenreider, "Alignment of Multiple Ontologies of Anatomy: Deriving Indirect Mappings from Direct Mappings to a Reference", AMIA 2005 Symposium Proceedings, 2005.
4. J. J. Jung, A. Hakansson, and R. H. . "Indirect Alignment between Multilingual Ontologies: A Case study of Korean and Swedish Ontologies," in Proceedings of the Third KES International Symposium on Agent and Multi-Agent Systems: Technologies and Applications,2009.
5. A. Khiat and M. Benaissa and Ernesto Jimnez-Ruiz "ADOM: arabic dataset for evaluating arabic and cross-lingual ontology alignment systems". In Proceedings of the 10th International Workshop on Ontology Matching co-located with the 14th International Semantic Web Conference (ISWC 2015), USA, 2015.
6. A. Khiat, G. Diallo, B. Yaman, E. Jimnez-Ruiz and M. Benaissa, "ABOM and ADOM: Arabic Datasets for the Ontology Alignment Evaluation Campaign". In Proceedings of the 14th International Conference (ODBASE 2015), Greece, 2015.
7. W. Djeddi, M. T.Khadir and S. Ben-Yahia, "XMap++ results for OAEI 2015". In Proceedings of the 10th International Workshop on Ontology Matching ISWC 2015, USA, 2015.
8. D. Faria, C. Martins, A. Nanavaty, D. Oliveira, B. Sowkarthiga, A. Taheri, C. Pesquita, F. Couto and I. Cruz , "AML results for OAEI 2015". In Proceedings of the 10th Workshop on Ontology Matching ISWC 2015, USA, 2015.
9. D. Ngo and Z. Bellahsene, "YAM++ results for OAEI 2013", In Proceedings of the 8th Workshop on Ontology Matching ISWC 2013, pp. 211-218, Australia, 2013.
10. A. Khiat and M. Benaissa, "AOT / AOTL results for OAEI 2014". In Proceedings of the 9th International Workshop on Ontology Matching ISWC 2014, pp. 113-119, Italy, 2014.
11. E. Jiménez-Ruiz, BC. Grau, A. Solimando, V. Cross, "LogMap family results for OAEI 2015". In Proceedings of the 10th Workshop on Ontology Matching ISWC 2015, USA, 2015.
12. O. Svab, V. Svatek, P. Berka, D. Rak and P. Tomasek, "OntoFarm: Towards an Experimental Collection of Parallel Ontologies", In: Poster Track of ISWC 2005, Galway, 2005.
13. C. Meilicke, R. Garca-Castro, F. Freitas, WR. Van Hage, E. Montiel-Ponsoda, R.R. De Azevedo, H. Stuckenschmidt, O. vb-Zamazal, V. Svtek and A. Tamilin, "MultiFarm: A benchmark for multilingual ontology matching". Web Semant. Sci. Serv. Agents World Wide Web. Vol. 15, pp. 62-68, 2012.
14. K. Kotis, A. Katasonov and J. Leino, "AUTOMSv2 results for OAEI 2012", In Proceedings of the 7th Workshop on Ontology Matching ISWC 2012, USA, 2012.
15. A. Gro, M. Hartung, T. Kirsten and E. Rahm, "GOMMA results for OAEI 2012", In Proceedings of the 7th Workshop on Ontology Matching ISWC 2012, USA, 2012.
16. H. Paulheim, "WeSeE-Match results for OEAI 2012", In Proceedings of the 7th Workshop on Ontology Matching ISWC 2012, USA, 2012.
17. S. Hertling and H. Paulheim, "WikiMatch results for OEAI 2012", In Proceedings of the 7th Workshop on Ontology Matching ISWC 2012, pp., USA, 2012.
18. M. El-Abdi, H. Souid, M. Kachroudi and S. Ben-Yahia, "CLONA results for OAEI 2015", In Proceedings of the 10th Workshop on Ontology Matching ISWC 2015, USA, 2015.
19. A. Khiat, M. Benaissa and M. A. Belfdhal, "STRIM results for OAEI 2015 instance matching evaluation". In Proceedings of the 10th International Workshop on Ontology Matching co-located with the 14th International Semantic Web Conference (ISWC 2015), USA, 2015.
20. A. Khiat, CroLOM: Cross-Lingual Ontology Matching System Results for OAEI 2016. In Proceedings of the 12th International Workshop on Ontology Matching co-located with the 15th International Semantic Web Conference (ISWC 2016), Japan, 2016.

I-Match and OntoIdea results for OAEI 2017

Abderrahmane Khiat and Maximilian Mackeprang

Human-Centered Computing Lab, Freie Universität Berlin, Germany
abderrahmane.khiat@fu-berlin.de, maximilian.mackeprang@fu-berlin.de

Abstract. Presenting a set of similar or diverse ideas during the idea generation process leads ideators to come-up with more creative and diverse ideas. However, to better assess the similarity between the ideas, we designed two matching systems, namely I-Match and OntoIdea. In the context of the idea generation process, each idea is represented by a set of instances from DBpedia describing the main concepts of the idea. Then, the developed matching systems are applied to compute the similarity between a set of instances that represent the ideas. The purpose of our participation at OAEI is to evaluate our designed instance matching algorithm in order to apply it to assess the similarity between ideas. The results obtained for the first participation of I-Match and OntoIdea systems at OAEI 2017, on different instance matching tracks are so far quite promising.

Keywords: Collaborative Ideation, Semantic Annotation, Ontology, Instance Matching, OAEI.

1 Introduction

The idea generation process is the key part of innovation. This process aims to generate ideas to solve problems and challenges. A promising approach for supporting such process is the ”brainstorming method” [3]. This method seeks to increase the number of ideas based on ideas of collaborating individuals while restricting criticism.

In addition to leveraging the crowd [10], prior work has shown that generating ideas that are both creative and diverse can be greatly enhanced through presenting inspirational examples [6]. However, a major issue is ”how to find inspiring ideas from hundreds” [9]. To overcome this challenge, research has shown three ways of selecting a set of inspiring examples systematically [4, 5]: (1) presenting diverse ideas, (2) presenting similar ideas and (3) visualizing all ideas.

Our work is in line with approaches that assess the diversity (i.e. low similarity rating) of inspiring examples automatically [8]. However, assessing similarity between ideas is challenging due to the form of the ideas, i.e. the ideas are described in a short unstructured text.

To solve this problem, we propose another strategy from our prior work proposed in [2]. This strategy consists of two main parts: (1) concepts annotation and (2) an instance matching mechanism. Firstly, we annotated the main concepts of an idea with instances from DBpedia, a validation through user-based selection of images are carried out in order to obtain the right meaning of the identified concepts. Secondly, these annotated concepts with a set of instances are used as a support to calculate the similarity between ideas using an instance matching system. Using our approach, we can

assess the similarity of two ideas, which can then be used further to select (1) a set of diverse ideas (low similarity rating), (2) a set of similar ideas (high similarity rating) that inspire the user to generate more creative ideas. Furthermore, we use the similarity ratings obtained to provide a visualisation of the solution space to give ideators an overview of the collaborative effort.

In this paper, we focus on the matching part of the proposed solution by describing our two instance matching systems I-Match and OntoIdea. The designed systems implement an enhancing algorithm that we proposed in our previous work [1]. The proposed algorithm extracts first all information about the two instances to be matched and normalizes them using NLP. Then, it applies edit distance as a matcher to calculate the similarities between the normalized information. Finally, the approach selects the equivalent instances based on the maximum of shared information between the two instances.

2 Instance Matching Algorithm

We summarize the algorithm of our developed systems to provide a general idea of the proposed solution. It consists of the following successive phases:

2.1 Extraction and Normalization

The system extracts from each individual Ii P_1 m_1; P_2 m_2,... a set of information m_1, m_2, ... using different properties P_1, P_2, ... Then, NLP techniques are applied to normalize these information. In particular, three pre-processing steps are performed: (1) case conversion (conversion of all words in same upper or lower case) (2) lemmatization stemming and (3) stop word elimination. Since String based algorithm is used to calculate the similarities between information, these steps are necessary.

2.2 Similarity Calculation

In this step, the system calculates the similarities between the normalized informations using edit distance as string matcher. Our system selects the maximum similarity values calculated between different informations by edit distance. If two informations are the same (based on maximum similarity values) the counter is incremented to 1, etc.

2.3 Identification

Finally, we apply a filter on maximum counter values in order to select the correspondences which mean that the selected correspondences (equivalent individuals) are those who share maximum informations.

3 Experimentation

The I-Match and Ontoidea systems participated only for instance matching tracks of OAEI 2017 evaluation campaign. For the results Please refer to the following website: `http://oaei.ontologymatching.org/2017/results/index.html`.

4 Conclusion

In this paper, we have introduced I-Match and OntoIdea, two systems specially designed to compute similarity between instances. The proposed algorithm is useful, especially when the instances contain terminological information. The developed systems provide a quite promising results, thus, we will be applied in the context of the idea generation process to asses similarity between ideas.

As future perspective, we attempt to apply enhance our instance matching algorithm especially for DORUMUS track.

Acknowledgements The authors acknowledge the financial support of the Federal Ministry of Education and Research of Germany in the framework of Ideas to Market (project number 03IO1617).

References

1. A. Khiat, M. Benaissa and M. A. Belfdhal, "STRIM results for OAEI 2015 instance matching evaluation". In Proceedings of the 10th International Workshop on Ontology Matching co-located with the 14th International Semantic Web Conference (ISWC 2015), USA, 2015.
2. A. Khiat, M. Mackeprang, C. Mller-Birn, "Semantic Annotation for Enhancing Collaborative Ideation", Semantics 2017, Netherlands.
3. Alex F. Osborn. 1963. Applied imagination; principles and procedures of creative problem-solving. Scribner, NY, 1963.
4. Pao Siangliulue, Kenneth Arnold, Krzysztof Gajos and Steven Dow. 2015. Toward collaborative ideation at scale-leveraging ideas from others to generate more creative and diverse ideas". In Proceedings of CSCW, 2015.
5. Pao Siangliulue, Joel Chan, Steven Dow and Krzysztof Gajos. 2016. IdeaHound: improving large-scale collaborative ideation with crowd-powered real-time semantic modeling. In Proceedings of UIST.
6. Richard Marsh, Joshua Landau and Jason Hicks. 1996. How examples may (and may not) constrain creativity. Memory and Cognition, 24 (5), pp. 669–680.
7. Nicholas Kohn and Steven Smith, 2011. Collaborative fixation: Effects of others' ideas on brainstorming. Applied Cognitive Psychology 25 (3), pp. 359–371.
8. Joel Chan, Pao Siangliulue, Denisa McDonald, Ruixue Liu, Reza Moradinezhad, Safa Aman, Erin Solovey, Krzysztof Gajos and Steven Dow. 2017. Semantically Far Inspirations Considered Harmful? Accounting for Cognitive States in Collaborative Ideation. In Proceedings of C& C'17.
9. Elahe Javadi, Judith Gebauer and Joseph Mahoney. 2013. The impact of user interface design on idea integration in electronic brainstorming: an attention based view. Journal of AIS 14, pp. 1–21.
10. Victor Girotto, Erin Walker and Winslow Burleson. 2017. The effect of peripheral micro-tasks on crowd ideation. In Proceedings of CHI '17, pp. 1843–1854.
11. Osvald Bjelland and Robert Wood. 2008. An inside view of IBM's' innovation jam'". MIT Sloan Management Review 50 (1), pp. 32–40.
12. Joel Chan, Steven Dang and Steven Dow. 2016. IdeaGens: enabling expert facilitation of crowd brainstorming. In Proceedings of CSCW, pp. 13–16.

OAEI 2017 results of KEPLER

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
LIPAH-LR 1 ES14, 2092, Tunis, Tunisie
{marouen.kachroudi,sadok.benyahia}@fst.rnu.tn
[2] BPH Center - INSERM U1219, Team ERIAS & LaBRI UMR5800,
Univ. Bordeaux
gayo.diallo@u-bordeaux.fr

Abstract. This paper presents and discusses the results produced by KEPLER for the 2017 Ontology Alignment Evaluation Initiative (OAEI 2017). This method is based on the exploitation of three different strategy levels. The proposed alignment method KEPLER is enhanced by the integration of powerful treatments inherited from other related domains, such as Information Retrieval (IR) [1]. For scaling, the method is equipped with a partitioning module. For the management of multilingualism, KEPLER develops a well-defined strategy based on the use of a translator, and this provides very encouraging results.

1 Presentation of the system

Given the substantial growth of the semantic Web users that create and update knowledge all over the world in a multitude of conceptualizations. This process has been accelerated due to a few initiatives which encourage all the active participants to make their data available to the public. These actors often publish their data sources in their own respective languages, in order to make this information interoperable and accessible to members of all communities [2]. As a solution, the ontology alignment process aims to provide semantic interoperable bridges between heterogeneous and distributed information systems. Indeed, the informative volume reachable via the semantic Web stresses needs of techniques guaranteeing the share, reuse and interaction of all resources [3]. The explicitation of the associated concepts related to a particular domain of interest resorts to ontologies, considered as the kernel of the semantic Web. In this register, KEPLER is an ontology alignment system dealing with the key challenges related to heterogeneous ontologies on the semantic Web, and it uses several hybrid alignment strategies. KEPLER is designed to discover alignments for both normal size and large scale ontologies. In addition, the proposed alignment approach has the ability to treat multilingual ontologies as well as monolingual ones.

1.1 State, purpose, general statement

The proposed method, KEPLER, exploits besides the classic techniques, an external resource, *i.e.*, a translator to deal with multilingualism. KEPLER implements an alignment strategy which aims at exploiting all the wealth of the used ontologies.

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 6 complementary modules as presented in figure 1.

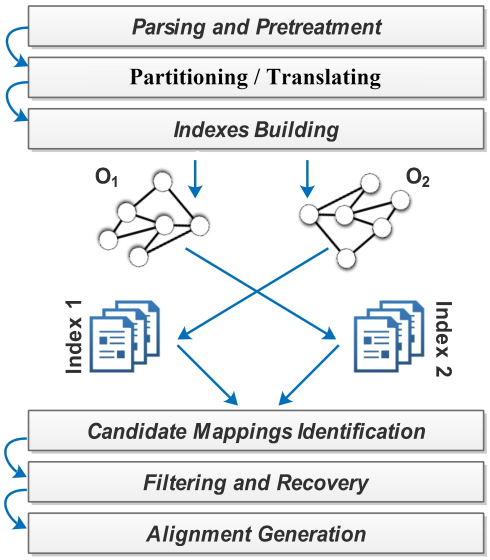


Fig. 1. KEPLER workflow.

Entities are described using OWL primitives with their semantics. We can then consider ontology as a semantic graph where entities are nodes connected by links which are OWL primitives. These links have specified semantic primitives. Consequently, if two ontologies in the same domain are similar, their semantic graphs are also the same.

Parsing and pretreatment This module allows to extract the ontological entities initially represented by a primary form of lists. In other words, at the parsing stage, we seek primarily to transform an OWL ontology in a well defined structure that preserves and highlight all the information contained in this ontology. Furthermore, in the resulting informative format, it has a considerable impact on the results of the similarity computation thereafter. Thus, we get couples formed by the entity name and its associated labels.

Partitioning This module aims at splitting ontologies into smaller parts to support the alignment task [4]. 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 both constituants : structures and semantics. One way to obtain such a

partitioning, can be to maximize the relationships inside a block and minimize the relationship between the blocks themselves. The partitioning quality result 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 can 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;
- *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 low outside.

Translation : An originality of our system, is to solve the heterogeneity problem mainly due to multilingualism, given the importance of this research area [5, 6]. This challenge brings us to choose between two alternatives, either we consider the translation path to one of the languages according to the two input ontologies, or we consider the translation path to a chosen pivot language. At this stage, we must have a foreseeable vision for the rest of our approach. Specifically, at the semantic alignment stage we use an external resource, *i.e.*, WordNet[3]. The latter is a lexical database for the English language. Therefore, the choice is governed by the use of WordNet, and we will prepare a translation of the two ontologies to the pivot language, which is English. To perform the translation phase we chose Bing Microsoft[4] tool.

Indexation : Indexing is one of the novelties of our approach. It consists in reducing the search space through the use of effective search strategy on the built indexes which represent the input ontologies components. To enable faster searching, the driving idea that was previously used in some works [1] is to execute the analysis 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 of the latter is activated. Thus, the query implementation satisfies the terminology search and semantic aspects at once as we are querying documents in a vector representation that contain a given ontological entity and its synonyms obtained via WordNet. It is worthy to mention that indexes querying is done in both senses.

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 aligned candidates. At this stage, once the list of candidates is ready, the alignment method uses the first filter. We should note that indexes querying may includes a set of redundant mappings. Doing so, this filter eliminates the redundancy. It goes through the list of candidates and for each candidate, it checks if there are duplicates. If this is the case, it removes the redundant element(s). At the end of filtering phase, we have a candidates list without redundancy, however, there is always the concern of *false positives*,

[3] https://wordnet.princeton.edu/
[4] https://www.bing.com/translator

in fact, there was the need to establish a second filter. Once the redundant candidates are deleted, the system uses the second filter that eliminates *false positives*. This filter is applied to what we call *partially* redundant entities. An entity is considered as *partially* redundant if it belongs to two different mappings (*i.e.*, being given three ontological entities e_1, e_2 and e_3. If on the one hand, e_1 is aligned to e_2, and secondly, e_1 is aligned to e_3, this last alignment is qualified as doubtful. We note that our method generates $(1 : 1)$ alignments. To overcome this challenge, the alignment method compares the topology of the 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, and retains the couple having the highest topological proximity value. All candidates are subject of this filter, and as output we have the final alignment file.

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 in the OAEI 2017.

2.1 Anatomy

This track consists of 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, and according to figure 2, KEPLER succeeded to extract 74% of correct mappings with a precision about 95%. Figure 2 summarizes the evaluation metrics values for Anatomy track. To this end, it is important to mention that KEPLER has managed to support the ontologies of the Anatomy database thanks to the *Ontopart* module [7, 4].

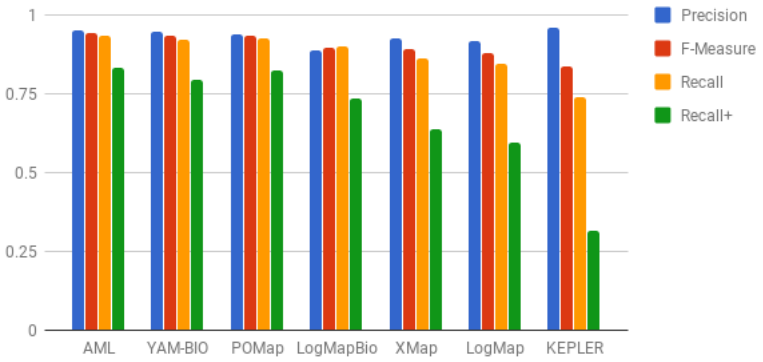


Fig. 2. KEPLER evaluation metrics among other pioneering systems for Anatomy track.

2.2 Conference

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 varies between 76% and 58%. Recall values varies between 48% and 68%. The metrics are obtained according to several evaluation scenarios.

2.3 Multifarm

This dataset is composed of a subset of the Conference track, translated in nine different languages (*i.e.*, Chinese, Czech, Dutch, French, German, Portuguese, Russian, Spanish and Arabic). 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 [8–13], 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.

KEPLER uses a specific technique to determine the equivalence between ontology entities described in different natural languages. We chose to use the English as a pivot language. The use of a pivot language ensures greater consistency of obtained translations since it starts from the same text. In the *different ontologies* case, the method is ranked fourth with a recall value of 0.31 as depicted by figure 3.

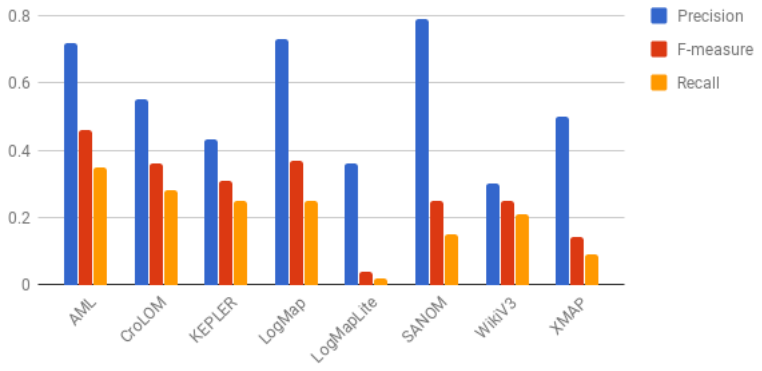


Fig. 3. KEPLER evaluation metrics among other pioneering systems for Multifarm track *(different ontologies)*.

Whereas in the *same ontologies* case, the method occupies the first place with a recall value of 0.52 as flagged by figure 4.

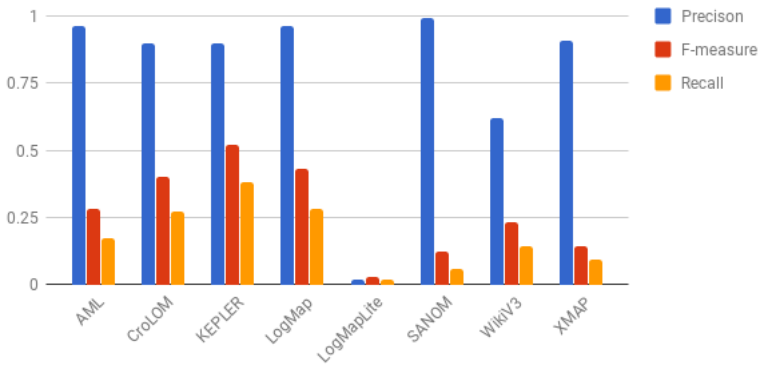


Fig. 4. KEPLER evaluation metrics among other pioneering systems for Multifarm track *(same ontologies)*.

2.4 Large Biomedical Ontologies and Phenotype

In the scalability register, this track consists of finding alignments between the Foundational Model of Anatomy (FMA), SNOMED CT, and the National Cancer Institute Thesaurus (NCI). These ontologies are semantically rich and contain tens of thousands of classes. 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 handles large ontologies in two phases: the first phase consists on partitioning the ontologies into a set of blocks and the second phase selects two suitable blocks giving the highest value of similarity to be aligned. KEPLER treated (*Task 1: FMA-NCI small fragments*)`[Precision : 0.96 / Recall : 0.83]` according to figure 5.

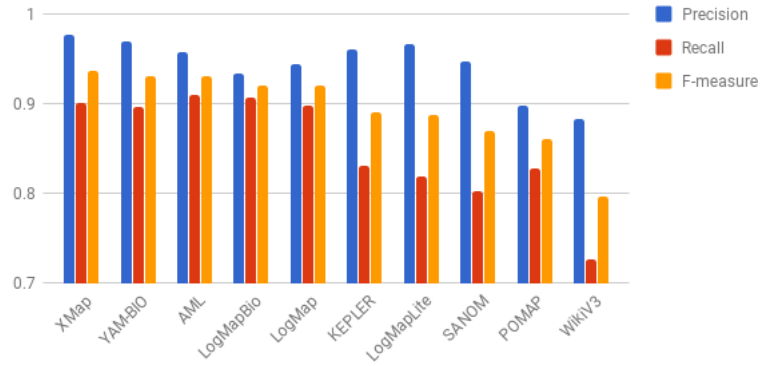


Fig. 5. KEPLER evaluation metrics among other pioneering systems for LargeBio track.

As depicted by figure 6, KEPLER processed also the task 3 of the LargeBio dataset (*FMA-SNOMED small fragments*) with a Precision of 0.82 and Recall of 0.55. In the Phenotype track, our method succeeds in processing only the DOID-ORDO sub-case by identifying 1824 matches for 1237 expected ones.

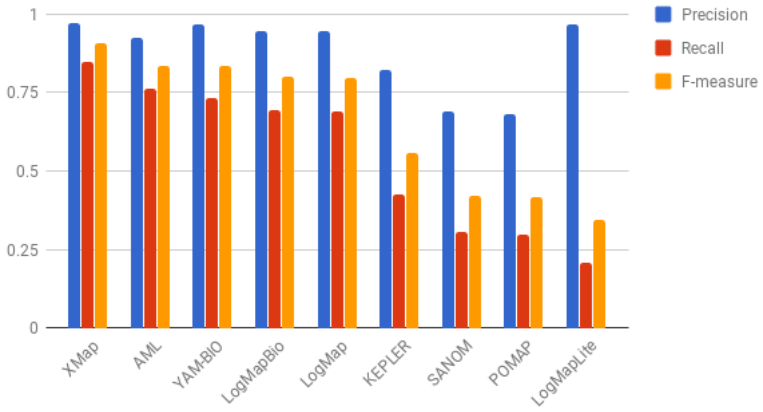


Fig. 6. KEPLER evaluation metrics among other pioneering systems for LargeBio track.

3 Conclusion

In this paper, we briefly presented the alignment system KEPLER with comments of the results obtained according to the OAEI 2017 tracks, corresponding to the SEALS platform evaluation modalities. 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 [12] and SERVOMAP [1]. It showed promising results for its first participation. As future work, we plan to consolidate our system to more support the instance based ontology alignment in a wider range and context. We have dealt with this issue before [14, 15], but the test base update imposes other challenges, in terms of the used ontological languages and the evolutive semantic description formalisms.

References

1. Diallo, G.: An effective method of large scale ontology matching. Journal of Biomedical Semantics **5(44) doi:10.1186/2041-1480-5-44** (2014)
2. 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
3. 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

4. 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
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. 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
8. 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
9. 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
10. 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
11. 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
12. 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
13. 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 Francophones sur les Ontologies (JFO'2016), Bordeaux, France (2016) 57–68
14. 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
15. Zghal, S., Kachroudi, M., Damak, S.: Alignement d'ontologies à base d'instances indexées. In: Actes de la 6$^{\text{ème}}$ Edition des Journées Francophones sur les Ontologies (JFO'2016), Bordeaux, France (2016) 69–74

Legato: Results for OAEI 2017

Manel Achichi, Zohra Bellahsene, Konstantin Todorov

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

Abstract. *Legato* is an automatic data linking system handling datasets containing blocks of highly similar in their descriptions but yet distinct resources, as well as resources with highly heterogeneous descriptions. This paper presents the results of *Legato* on the Instance Matching track of the Ontology Alignment Evaluation Initiative 2017 via the SEALS platforme. *Legato* participated in the two sub-tracks of the instance matching track. We briefly describe the *Legato* framework, we present the different techniques used by the system in the accomplishment of the data linking task and we present and discuss the alignment results of the system as compared to the other tools participating to the 2017-edition of the evaluation campaign.

1 Presentation of the System

We begin by providing an overview of the main characteristics of *Legato*, as well as describing briefly the specific techniques applied in the different parts of its workflow.

1.1 General Features and Purpose

Legato is a data linking tool developed in the framework of the DOREMUS project[1]. It is designed to match entities from highly heterogeneous graphs, effectively disambiguating highly similar (yet distinct) resources. *Legato* is based on indexing techniques, with a preliminary phase of data cleaning allowing to prune properties that make the comparison task difficult, as well as a post-processing phase allowing to discard erroneous links and to lower the rate of false positives. An important feature of our system is that it requires very little manual configuration – neither similarity measures and thresholds, nor properties to align are required as input. The values of the various thresholds inherent to the algorithm are set empirically so as to ensure a maximum performance on a large variety of heterogeneous data. With this, we aim at placing *Legato* among the few fully automatic instance matchers in the state of the art. The system is openly available at the following link: `https://github.com/DOREMUS-ANR/legato`.

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

1.2 Specific Techniques Used

This section briefly describes the overall workflow of *Legato*, shown in Figure 1. Its configuration takes one single parameter: the type of resources for comparing and linking. The system then proceeds to automatically process, compare, repair and provide a set of identity links (`owl:sameAs` statements). More precisely, *Legato* implements the following successive steps.

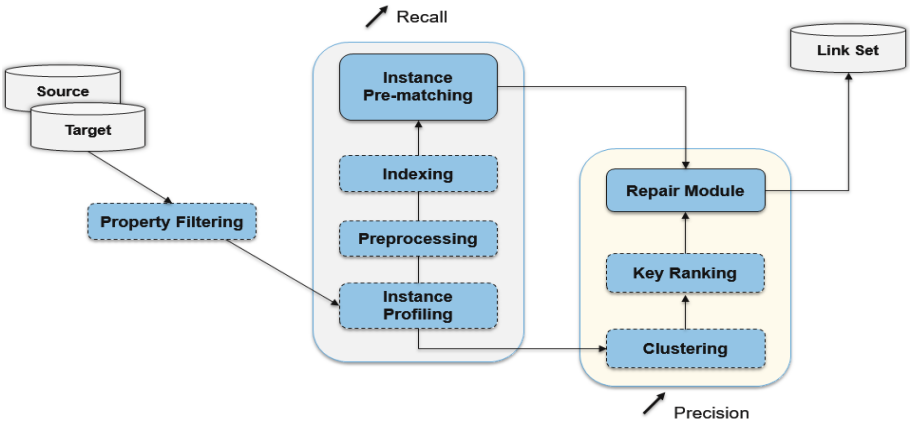


Fig. 1: The processing pipeline of *Legato*

Data cleaning. The first step before representing the resources in a comparable form consists in filtering the problematic properties from the two input datasets. *Legato* considers a property as *problematic* if it hinders the comparison of resources. Consider the example given in Table 1, issued from the DOREMUS track data from the IM@OAEI2017 (Instance Matching track of the Ontology Alignment Evaluation Initiative).

The descriptions `mw1` and `mw1'` are about two equivalent musical works retrieved from *Philharmonie de Paris* (PP) and *Bibliothèque Nationale de France* (BNF), respectively. These descriptions are highly similar, with the notable exception of the respective `ecrm:P3_has_note` property values. Considering this property, we would yield a very low value of the similarity score, and still it is likely that this property is discovered as a key (because of its unique values) and therefore used in a configuration file of a linking system.

Properties identified as problematic may concern those that have values in a free text format, i.e., comments (as in the example above), as well as resource-specific values, that the publisher cannot describe freely. For example, for the same musical work, two institutions would generally assign different identifiers in their respective catalogs. The way we propose to identify automatically problematic properties, is to discover mono-property keys valid **on both** datasets, i.e., each object for such a property has at most one subject in both datasets.

`mw1`[2]	a efrbroo:F22_SelfContained_Expression mus:U70_has_title "Sonates" mus:U12_has_genre sonate[3] ecrm:P3_has_note "Cette sonate est constituée de cinq formants: Antiphonie, Trope, Constellation, Strophe et Séquence. Seuls les 2e et 3e formants sont publiés. Le Formant 2 (Trope) est composé de quatre sections : Commentaire, Glose , Texte, Parenthèse, qui peuvent être jouées dans différents ordres. Cette oeuvre nécessite un piano à 3 pédales. - Durée d'exécution : 20 minutes environ"
`mw1'`[4]	a efrbroo:F22_SelfContained_Expression mus:U70_has_title "Sonates" mus:U12_has_genre sonate[5] ecrm:P3_has_note "Date de révision : 1963, comprend : Antiphonie; Trope; Constellation (ou Constellation-Miroir); Strophe; Séquence"

Table 1: `ecrm:P3_has_note` — An example of a problematic property in DOREMUS data

Instance profiling. *Legato* creates instance profiles by exploiting the information in the $\mathcal{CBD}$s (for Concise Bounded Description) of the resources.[6] We extend the $\mathcal{CBD}$ notion by also considering the descriptions of neighboring nodes of a resource in its graph. At this step, *Legato* extracts a subgraph for each resource `r` that includes all the triples from the $\mathcal{CBD}$ of `r`, the $\mathcal{CBD}$s of its direct predecessors (linked by incoming links to `r`), and the $\mathcal{CBD}$s of its direct successors (linked through outgoing links to `r`). For instance profiling, *Legato* only considers datatype properties. In that, each resource is represented by a set of literals in its profile (subgraph) considered as relevant for its description. This strategy allows to avoid manually setting the graph traversal distance to which the information should be collected.

Instance pre-matching. Once all resources in both datasets are profiled, *Legato* employs an indexing technique to project each profile onto a vector space where terms are weighted by their TF-IDF (Term Frequency-Inverse Document Frequency) values. Two standard NLP (Natural Language Processing) filters are applied: tokenization and stop-words removal. Finally, *Legato* pre-selects the identity links by computing the correlation between vectors by using the well-known cosine similarity. In order to increase recall and to automate the threshold setting independently on the data, at this stage *Legato* generates links with a very low threshold (empirically fixed at 0.2).

Link repairing. To ensure coherence, the alignments selected at the pre-matching step are passed to the *repair* module. Note that decreasing the similarity threshold may increase the number of false positive matches. As indicated above, a source resource may be erroneously aligned to many target resources

[6] `https://www.w3.org/Submission/CBD/`

(and vice versa). This is due to the fact that we can have highly similar descriptions of different resources in a single dataset. Therefore, *Legato* includes a post-processing phase allowing to disambiguate between such resources and to repair the erroneous links generated between them in the previous phase. We employ a clustering algorithm [1] within each dataset aiming to group together the similar resources. Then, for each pair of similar clusters (identified by a cluster matching algorithm) across the two datasets, the resources are compared on a best-key basis. We apply the RANKey algorithm for identifying and ranking the key properties [2]. For each link $\mathtt{l}=(\mathtt{r_s}, \mathtt{r_t})$ produced in the earlier step, the repair module begins by searching for a link of $\mathtt{r_s}$ to a target resource $\mathtt{r'_t} \neq \mathtt{r_t}$, based on the key strategy. If found, the target resource $\mathtt{r_t}$ in $\mathtt{l}$ is then replaced by $\mathtt{r'_t}$. In case multiple matches are found in that scenario, the one with the highest similarity score is kept. The repair module aims at improving precision.

Link to the System and Parameters File. We provide an open source implementation of *Legato* in a GitHub project under the following link: `https://github.com/DOREMUS-ANR/legato`. It is available as an eclipse project. *Legato* provides an appropriate user interface allowing the user to select the source, target and alignment (if it is available) files for aligning and evaluating the produced links. If no alignment file exists, *Legato* produces a set of identity links without evaluating them.

Link to the Set of Provided Alignments. The alignments produced by *Legato* on the instance matching track of OAEI2017 can be downloaded at `https://github.com/manoach/Legato-at-OAEI-2017`.

2 Results

In this section, we present the results obtained by *Legato* on the data coming from the instance matching track of the OAEI2017 campaign.[7] This year, the instance matching track contains two tasks and four datasets. *Legato* participated to all these tasks.

2.1 Synthetic Task

This task contains synthetic data about creative works. They have been generated through the Semantic Publishing Instance Matching Benchmark (SPIMBENCH) [3] by transforming the source instances based on their values, structure and semantics. The task contains two matching sub-tasks on two different datasets: SPIMBENCH sandbox and SPIMBENCH mainbox (datasets of different sizes). The first one contains 380 resources while the second one – 1800.

Tables 2 and 3 show *Legato*'s results as compared to those of the other systems that have participated at this task, namely, AML, I-Match and LogMap. As it

[7] `http://oaei.ontologymatching.org/2017/`

System	Precision	Recall	F-measure
AML	0.849	**1.000**	0.918
I-Match	0.854	0.997	**0.920**
Legato	**0.980**	0.730	0.840
LogMap	0.938	0.763	0.841

Table 2: Results for SPIMBENCH sandbox.

System	Precision	Recall	F-measure
AML	0.855	**1.000**	**0.922**
I-Match	0.856	0.997	0.921
Legato	**0.970**	0.700	0.810
LogMap	0.893	0.709	0.790

Table 3: Results for SPIMBENCH mainbox.

can be seen, *Legato* achieves the highest score in terms of precision for both SPIMBENCH sandbox and SPIMBENCH mainbox (98% and 97%, respectively). We notice that Legato performs overall well on this task achieving a recall of 73% and 70%, and F-measures of 84% and 81% for SPIMBENCH sandbox and SPIMBENCH mainbox, respectively.

2.2 DOREMUS Task

The data from the DOREMUS track contain descriptions of real-world classical music works and events, coming from the catalogs of two major French cultural institutions (the Philharmonie de Paris and the National Library). These data have been converted to RDF from their original MARC format by the help the specifically designed for that purpose by the DOREMUS team tool *marc2rdf*.[8] These data follow a common ontology [4] given by the DOREMUS model, extending well-established models for intellectual works description, historically used by libraries.[9]

System	Precision	Recall	F-measure
AML	0.851	0.479	0.613
I-Match	0.680	0.071	0.129
Legato	0.930	0.920	0.930
LogMap	0.406	0.882	0.556
NjuLink	**0.966**	**0.945**	**0.955**

Table 4: Results for HT of the DOREMUS task

System	Precision	Recall	F-measure
AML	0.914	0.427	0.582
I-Match	**1.000**	0.053	0.101
Legato	**1.000**	**0.980**	**0.990**
LogMap	0.119	0.880	0.210
NjuLink	0.959	0.933	0.946

Table 5: Results for FPT of the DOREMUS task

Tables 4 and 5 show *Legato*'s results and those of the four other systems that participated at this task, namely, AML, I-Match, LogMap and NjuLink.

[8] `https://github.com/DOREMUS-ANR/marc2rdf`
[9] `http://data.doremus.org/ontology/`

On both subtasks, two systems stand out in terms of performance – *Legato* and NjuLink, achieving comparable results and outperforming considerably the other participant systems. More precisely, on the Heterogeneities task (HT data), *Legato* ranks second after NjuLink with a precision of 93%, a recall of 92% and F-measure of 93%. As for the False Positives Trap task (FTP data), it can be seen in Table 5 that *Legato* achieves the best results in terms of precision (100%), recall (98%) and F-measure (99%). It is worth noting that the DOREMUS track appeared to be problematic for the majority of the systems, with average F-measure scores of around 0.6 over all participants on both tasks.

3 Discussion

As seen in the previous section, our system proves to be very effective for the two sub-tracks of the instance matching track of OAEI 2017, showing its strength of producing high scores in terms of F-measure (above 80% on all tasks). *Legato* produced the best precision in 3 of the 4 instance matching tasks. Thanks to its repair module, *Legato* ensures a very high accuracy, which is no less than 93% on all instance matching tasks. In terms of recall, *Legato* scored well on the DOREMUS track, but obtained the lowest rank on the synthetic data track. We explain that result by the fact that *Legato* does not yet tackle value-based variations that are characteristic for the synthetic data – the lack of lemmatization in the indexing process of our system equates to looking only for exact matches between string values.

Proposed Improvements of the System *Legato* implements an approach handling structurally heterogeneous descriptions. However, the limit of the current version of our system is that it is not dealing with value-based heterogeneity, but rather considers exact matches only. Therefore, this will be the main base of future improvements. Furthermore, we plan to discover matches between resources coming from multiple data sources simultaneously.

4 Conclusion

In this paper, we presented *Legato*—an automatic and generic data linking tool. *Legato* participates for the first time at the OAEI campaign and it was evaluated on data from the two sub-tracks of the Instance Matching track. The results showed that *Legato* is capable of effectively linking both synthetic and real-world data of highly heterogeneous nature achieving comparable results to the best systems and outperforming most of them in terms of precision while keeping a decent recall level. In addition, *Legato* achieved the best score on the FPT DOREMUS data containing highly similar resources, thanks to its post-processing link repairing step. Finally, *Legato* is among the few participant systems that are freely available and ready to use by researchers or practitioners.

Acknowledgements

This work has been partially supported by the French National Research Agency(ANR) within the DOREMUS Project, under grant number ANR-14-CE24-0020.

References

1. L. Rokach and O. Maimon, "Clustering methods," in *The Data Mining and Knowledge Discovery Handbook.*, pp. 321–352, 2005.
2. M. Achichi, M. Ben Ellefi, D. Symeonidou, and K. Todorov, "Automatic key selection for data linking," in *Knowledge Engineering and Knowledge Management: 20th International Conference, EKAW 2016, Bologna, Italy, November 19-23, 2016, Proceedings 20*, pp. 3–18, Springer, 2016.
3. T. Saveta, E. Daskalaki, G. Flouris, I. Fundulaki, M. Herschel, and A.-C. 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*, pp. 105–106, ACM, 2015.
4. M. Achichi, R. Bailly, C. Cecconi, M. Destandau, K. Todorov, and R. Troncy, "Doremus: Doing reusable musical data," in *ISWC: International Semantic Web Conference*, 2015.

LogMap family participation in the OAEI 2017

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

[1] Department of Informatics, University of Oslo, Oslo, Norway
[2] Department of Computer Science, University of Oxford, Oxford, UK
[3] Computer Science and Software Engineering, Miami University, Oxford, OH, United States

Abstract. We present the participation of LogMap and its variants in the OAEI 2017 campaign. The LogMap project started in January 2011 with the objective of developing a scalable and logic-based ontology matching system. This is our seventh 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

Ontology matching systems typically rely on lexical and structural heuristics and the integration of the input ontologies and the mappings may lead to many undesired logical consequences. In [13] three principles were proposed to minimize the number of potentially unintended consequences, namely: *(i) consistency principle*, the mappings should not lead to unsatisfiable classes in the integrated ontology; *(ii) locality principle*, the mappings should link entities that have similar *neighbourhoods*; *(iii) conservativity principle*, the mappings should not introduce alterations in the classification of the input ontologies. Violations to these principles may hinder the usefulness of ontology mappings. The practical effect of these violations, however, is clearly evident when ontology alignments are involved in complex tasks such as query answering [22].

LogMap [12, 14] is a highly scalable ontology matching system that implements the consistency and locality principles. LogMap also supports (real-time) user interaction during the matching process, which is essential for use cases requiring very accurate mappings. LogMap is one of the few ontology matching system that *(i)* can efficiently match semantically rich ontologies containing tens (and even hundreds) of thousands of classes, *(ii)* incorporates sophisticated reasoning and repair techniques to minimise the number of logical inconsistencies, and *(iii)* provides support for user intervention during the matching process.

LogMap relies on the following elements, which are keys to its favourable scalability behaviour (see [12, 14] for details).

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

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, 23]. 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 2017 campaign

As in previous campaigns, in the OAEI 2017 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[4].

1.2 Adaptations made for the 2017 evaluation

LogMap's algorithm described in [12, 14, 16, 15] has been adapted with the following new functionalities:

i **Extended instance matching support.** We have adapted LogMap's instance matching module to be more flexible and adaptable to new matching tasks.

ii **Overlapping estimation.** We have also slighly improved the overlapping estimation module to reduce the search space. It now considers an extended set of labels necessary to apply the overlapping estimation in the new datasets of the Disease & Phenotype track [11].

[4] LogMapC is a variant of LogMap which, in addition to the consistency and locality principles, also implements the conservativity principle (see details in [24–26, 19]).

iii **Extended interactive support.** The interactive algorithm makes now use of the functionalities of the SEALS client and allows to make several related questions in one go.

1.3 Link to the system and parameters file

LogMap is open-source and released under GNU Lesser General Public License 3.0.[5] LogMap components and source code are available from the LogMap's GitHub page: `https://github.com/ernestojimenezruiz/logmap-matcher/`.

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

LogMap, including support for interactive ontology matching, can also be used directly through an AJAX-based Web interface: `http://krrwebtools.cs.ox.ac.uk/`. This interface has been very well received by the community since it was deployed in 2012. More than 2,800 requests coming from a broad range of users have been processed so far.

1.4 Modular support for mapping repair

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

We believe that these systems could significantly improve their output if they were to implement repair techniques similar to those available in LogMap. Therefore, with the goal of providing a useful service to the community, we have made LogMap's ontology repair module (LogMap-Repair) available as a self-contained software component that can be seamlessly integrated in most existing ontology matching systems [18, 7].

2 General comments and conclusions

Please refer to `http://oaei.ontologymatching.org/2017/results/` for the results of the LogMap family in the OAEI 2017 campaign.

2.1 Comments on the results

LogMap has been one of the top systems in the OAEI 2017 and one of the few systems that participates in (almost) all tracks.[6] 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.

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

[6] Participates in all SEALS tracks, but does not participate in the HOBBIT track.

2.2 Discussions on the way to improve the proposed system

LogMap is now a stable and mature system that has been made available to the community and has been extensively tested. There are, however, many exciting possibilities for future work. For example we aim at improving the current multilingual features and the current use of external resources like BioPortal. Furthremore, we are applying LogMap in practice in the domain of oil and gas industry [21, 17, 10, 20]. This practical application presents a very challenging problem.

Acknowledgements

This work was partially funded by the BIGMED project (IKT 259055), the HealthInsight project (IKT 247784), 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. Giese, M., Soylu, A., Vega-Gorgojo, G., Waaler, A., Haase, P., Jimenez-Ruiz, E., Lanti, D., Rezk, M., Xiao, G., Ozcep, O., Rosati, R.: Optique — Zooming In on Big Data Access. Computer 48(3), 60–67 (2015)
11. Harrow, I., Jiménez-Ruiz, E., Splendiani, A., Romacker, M., Woollard, P., Markel, S., Alam-Faruque, Y., Koch, M., Malone, J., Waaler, A.: Matching disease and phenotype ontologies in the ontology alignment evaluation initiative. Journal of Biomedical Semantics 8(1) (Dec 2017), https://doi.org/10.1186/s13326-017-0162-9
12. Jiménez-Ruiz, E., Cuenca Grau, B.: LogMap: Logic-based and Scalable Ontology Matching. In: Int'l Sem. Web Conf. (ISWC). pp. 273–288 (2011)

13. Jiménez-Ruiz, E., Cuenca Grau, B., Horrocks, I., Berlanga, R.: Logic-based assessment of the compatibility of UMLS ontology sources. J. Biomed. Sem. 2 (2011)
14. Jiménez-Ruiz, E., Cuenca Grau, B., Zhou, Y., Horrocks, I.: Large-scale interactive ontology matching: Algorithms and implementation. In: Europ. Conf. on Artif. Intell. (ECAI) (2012)
15. Jiménez-Ruiz, E., 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., Kharlamov, E., Zheleznyakov, D., Horrocks, I., Pinkel, C., Skjæveland, M.G., Thorstensen, E., Mora, J.: BootOX: Practical Mapping of RDBs to OWL 2. In: International Semantic Web Conference (ISWC) (2015), http://www.cs.ox.ac.uk/isg/tools/BootOX/
18. Jiménez-Ruiz, E., Meilicke, C., Cuenca Grau, B., Horrocks, I.: Evaluating mapping repair systems with large biomedical ontologies. In: 26th Description Logics Workshop (2013)
19. 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)
20. Kharlamov, E., Hovland, D., Jiménez-Ruiz, E., Lanti, D., Lie, H., Pinkel, C., Rezk, M., Skjæveland, M.G., Thorstensen, E., Xiao, G., Zheleznyakov, D., Horrocks, I.: Ontology Based Access to Exploration Data at Statoil. In: International Semantic Web Conference (ISWC). pp. 93–112 (2015)
21. Kharlamov, E., Jiménez-Ruiz, E., Zheleznyakov, D., et al.: Optique: Towards OBDA Systems for Industry. In: Eur. Sem. Web Conf. (ESWC) Satellite Events. pp. 125–140 (2013)
22. Meilicke, C.: Alignment Incoherence in Ontology Matching. Ph.D. thesis, University of Mannheim (2011)
23. Nebot, V., Berlanga, R.: Efficient retrieval of ontology fragments using an interval labeling scheme. Inf. Sci. 179(24), 4151–4173 (2009)
24. 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)
25. 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)
26. 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/

njuLink: Results for Instance Matching at OAEI 2017

Xinze Lyu, Qingheng Zhang, Wei Hu(✉), Zequn Sun, and Yuzhong Qu

[1] State Key Laboratory for Novel Software Technology, Nanjing University, China
[2] Department of Computer Science and Technology, Nanjing University, China
{xzlv.nju,qhzhang.nju,zqsun.nju}@gmail.com,{whu,yzqu}@nju.edu.cn

Abstract. njuLink is a tool designed for instance matching. It mainly matches instances by finding discriminative property pairs. Also, to meet 1:1 equivalence relationship for the OAEI 2017 DORUMES task, we make several improvements. In this report, we describe the design ideas and show our evaluation results.

1 Presentation of the System

1.1 State, purpose, general statement

With the rapid development of the Semantic Web, the amount of RDF data on the Semantic Web is growing in an unprecedented pace. This also brings great challenges to instance matching. On the Semantic Web, an instance describes a real-world object, it is composed of a subject and many $\langle p, v\rangle$ pairs, where p denotes a "property" and v denotes a "value". Subject serves as unique token for a real-world object, and $\langle p, v\rangle$ pairs describe the features of this real-world object. Instance matching aims to find the instances that describe the same real-world object and establish links between them. If two instances describe the same real-world object, we consider them as *coreferent instances* or a *coreferent instance pair*. Thanks to a lot of existing work, e.g., the Linked Open Data (LOD) Initiative, millions of links have been established. But, there are still a huge number of instances that potentially refer to the same object but have not been interlinked yet.

Our previous work tries to find coreferent instances by discriminative properties [2]. This approach is very effective but needs some improvements to meet the requirements of the DOREMUS task, which is to find 1:1 equivalence relationship between two datasets. So, we design njuLink, where "nju" represents "Nanjing University". The key idea of njuLink lies in finding what is essential to determine whether two instances are coreferent. Driven by this, first, njuLink builds a small-scale training set via predicting coreferent and non-coreferent instance pairs. Then, by analyzing the value similarity of every instance pair in training set, njuLink finds some property pairs named *discriminative property pairs*, which have the ability to identify whether two instances are coreferent. Finally, for an instance pair, njuLink calculates the similarity of values based

on the discriminative property pairs, the similarity of values based on common property pairs and the similarity of properties that they have to determine if the instances in this pair is coreferent.

1.2 Specific techniques used

There are four steps in the workflow of njuLink, which is shown in Fig. 1. We will describe the strategies to calculate the similarity of values and the similarity of properties shortly.

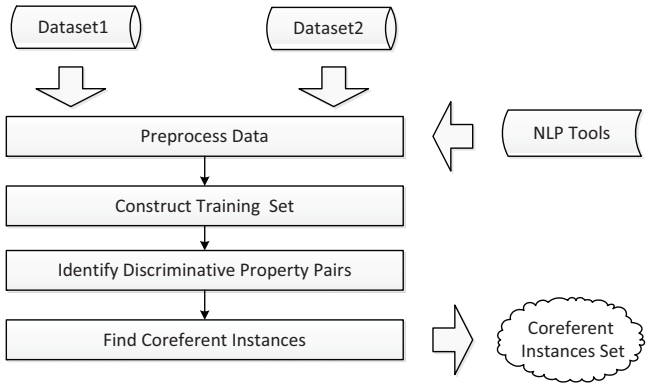


Fig. 1. The work flow of njuLink

The task we participated in is to find coreferent instance pairs between two datasets. To make our descriptions more clear, we give some notations as follows: (1) Let D^x and D^y be two different datasets, respectively; (2) The elements with superscript x are from D^x and those with superscript y are from D^y, e.g., instances, properties and values in D^x are i^x, p^x and v^x, respectively; and (3) Every instance pair $\langle i^x, i^y \rangle$ mentioned in this article is composed of an instance i^x from D^x and an instance i^y from D^y, and i^x is written to the left and i^y is written to the right, this also applies to property pairs $\langle p^x, p^y \rangle$ and value pairs $\langle v^x, v^y \rangle$.

Preprocess Data. For an instance, njuLink preprocesses the values describing it. There are three types of values: Blank node, URI and Literal (plain or typed). If a value is blank node, njuLink ignores it. Literal is divided into two kinds: typed literal, like boolean and integer, and plain literal, which is often accompanied with a language tag.

First, njuLink records the type of each value. Then, if the value has a language tag, njuLink also records it. Thirdly, for literals, njuLink replaces punctuations and stop words like “at”, “in”, “for” with space by a NLP tool, and then njuLink removes all space. For URIs, njuLink only records its local name. Finally, njuLink

transforms subjects, properties and values to lowercase letters and stores them for the next step.

Strategies to Calculate Similarity. We describe our strategies to obtain the similarity of a value pair and the similarity of a property pair next.

Calculate similarity of a value pair. Let v^x and v^y be two values owned by properties p^x and p^y, respectively. First, njuLink judges whether v^x and v^y are meaningful to be compared. There are three situations under which comparing them are not meaningful: (1) They both have language tags and their language tags are different; (2) The types of them are different; and (3) One of them is blank node.

Second, let $T(v^x)$ be the type of v^x. If v^x and v^y are meaningful to be compared, the strategies to find their similarity, denoted by $ValSim(v^x, v^y \mid p^x, p^y)$, vary with their types:

$$ValSim(v^x, v^y \mid p^x, p^y) = \begin{cases} indicatorFunc(v^x, v^y), & T(v^x) = \text{typed literal} \\ \text{I-Sub}(v^x, v^y), & \text{otherwise} \end{cases} \quad (1)$$

where for typed literal, njuLink uses indicator function ($indicatorFunc(v^x, v^y)$) to get their similarity, e.g., when two literals are both date time type, their similarity is 1 if the two literals are equal, and 0 otherwise. For URI and plain literal, njuLink uses I-Sub [3] to calculate the similarity. When the similarity of v^x and v^y is higher than a threshold, they are considered as a similar value pair. The threshold is set to 0.65, which is suggested by the authors of I-Sub [3].

Calculate similarity of a property pair. Let p^x and p^y be two properties owned by instances i^x and i^y, respectively. A property may have more than one value, we let the sets of values of p^x and p^y be $Val(p^x, i^x)$ and $Val(p^y, i^y)$, respectively. First, we find value set that has a smaller size. Without loss of generality, we assume that $Val(p^x, i^x)$ is the smaller one here. For a value v^x in $Val(p^x, i^x)$, the maximum similarity between it and the values in $Val(p^y, i^y)$ is calculated by $MaxValSim(v^x, Val(p^y, i^y))$. The maximum similarity between values of $Val(p^x, i^x)$ and $Val(p^y, i^y)$, which is also considered as the maximum similarity of property pair $\langle p^x, p^y \rangle$, is denoted by $MaxPropSim(p^x, p^y \mid i^x, i^y)$:

$$MaxValSim(v^x, Val(p^y, i^y)) = \max_{v_n^y \in Val(p^y, i^y)} ValSim(v^x, v_n^y \mid p^x, p^y), \quad (2)$$

$$MaxPropSim(p^x, p^y \mid i^x, i^y) = \max_{\substack{v_m^x \\ \in Val(p^x, i^x)}} MaxValSim(v_m^x, Val(p^y, i^y)). \quad (3)$$

If $MaxValSim(v^x, Val(p^x, i^x))$ of v^x is higher than a threshold (i.e. 0.65), value v^x is considered as a matched value, we define the sets of matched values and unmatched values between p^x of i^x and p^y of i^y as follows:

$$MatVal(p^x, p^y \mid i^x, i^y) = \{v \mid v \in Val(p^x, i^x) \cap MaxValSim(v, Val(p^y, i^y)) > 0.65)\}, \quad (4)$$

$$UnmatVal(p^x, p^y \mid i^x, i^y) = \{v \mid v \in Val(p^x, i^x) \cap v \notin MatVal(p^x, p^y \mid i^x, i^y)\}. \quad (5)$$

If $MaxPropSim(p^x, p^y \mid i^x, i^y)$ is higher than a threshold (0.65), the property pair $\langle p^x, p^y \rangle$ is similar w.r.t. instance pair $\langle i^x, i^y \rangle$. Note that this property pair is not guaranteed to be similar in another instance pair. For every matched value v^x of $Val(p^x, i^x)$, we sum up its similarity by $MatValSimSum(p^x, p^y \mid i^x, i^y)$:

$$MatValSimSum(p^x, p^y \mid i^x, i^y) = \sum_{\substack{v_m^x \in MatVal(\\ p^x, p^y \mid i^x, i^y)}} MaxValSim(v_m^x, Val(p^y, i^y)). \tag{6}$$

Construct Training Set. Let D^x and D^y be two different datasets and $\langle i_m^x, i_n^y \rangle$ be an instance pair, where i_m^x is from D^x and i_n^y is from D^y. The training set is divided into two parts, Positives and Negatives. Positives consist of coreferent instance pairs and Negatives are composed of non-coreferent instance pairs.

To construct Positives, njuLink picks up 20 instance pairs that have at least one property pair whose maximum similarity is very high. The threshold of similarity under this situation is 1.

When it comes to Negatives, njuLink chooses 20 instances from D^y randomly to form an instance set, namely instSet^y. These 20 instances should be under the same class of i_n^y in Positives, i.e., if instances in Positives are to describe "student", the instances selected should describe "student", too.

Then, njuLink picks up instances i_m^x from every instance pair $\langle i_m^x, i_n^y \rangle$ in Positives to form another instance set, namely instSet^x. So, instSet^x contains 20 instances because there are 20 instance pairs in Positives. After that, for every one in instSet^x, njuLink selects an instance from instSet^y and makes them an instance pair. Note that every instance in instSet^x and instSet^y is used only once. Finally, 20 generated instance pairs constitute the Negatives.

These 20 generated instance pairs can be considered as non-conferent ones approximately because the number of non-coreferent instances is much more than that of coreferent instances and njuLink constitutes instSet^y by selecting instances randomly.

Identify Discriminative Property Pairs. For every instance pair $\langle i_m^x, i_n^y \rangle$ from Positives, where i_m^x and i_n^y represent two different instances, njuLink makes every property of i_m^x and every property of i_n^y a pair. Then, njuLink finds out which property pair is similar and records it. So, njuLink can get the frequency of every similar property pair recorded after checking all instance pairs. If the frequency of a property pair is more than half of the size of Positives, which equals 10 in this case, njuLink records it in candidate property pair set.

For every property pair $\langle p_k^x, p_j^y \rangle$ in candidate property pair set, where p_k^x and p_j^y represent properties, njuLink calculates the maximum similarity that an instance pair $\langle i^x, i^y \rangle$ on it ($MaxPropSim(p_k^x, p_j^y \mid i^x, i^y)$). If the similarity is higher than a threshold, which is 0.65, this instance pair is a coreferent instance pair found by $\langle p_k^x, p_j^y \rangle$, otherwise, this instance pair is not coreferent judged by $\langle p_k^x, p_j^y \rangle$.

The percentage of the number of coreferent instances found can measure the discriminability of a property pair, but we found a better approach in [1] to use *information gain*, which is widely used in classification.

Every property pair $\langle p_k^x, p_j^y \rangle$ of candidate property pair set can classify the whole training set to four sets, TP, FP, TN and FN, which denote true positives, false positives, true negatives and false negatives respectively. When an instance pair is coreferent, if it is also a coreferent one found by $\langle p_k^x, p_j^y \rangle$, it is put into TP, otherwise, it is put into FP. When an instance pair is not coreferent, if it is also a non-coreferent one judged by $\langle p_k^x, p_j^y \rangle$, it is put into TN, otherwise, it is put into FN.

Finally, let T be the training set, which is the union of Positives (T^+) and Negatives (T^-). For every property pair $\langle p_k^x, p_j^y \rangle$ in candidate property pair set, njuLink uses four sets generated by it to obtain the *information gain* of it, denoted by $IG(p_k^x, p_j^y)$:

$$IG(p_k^x, p_j^y) = E(T) - E(T_{\langle p_k^x, p_j^y \rangle}), \tag{7}$$

$$E(T) = \frac{|T^+|}{|T|} \log \frac{|T^+|}{|T|} - \frac{|T^-|}{|T|} \log \frac{|T^-|}{|T|}, \tag{8}$$

$$E(T_{\langle p_k^x, p_j^y \rangle}) = \frac{|P|}{|T|} E(P) - \frac{|Q|}{|T|} E(Q), \tag{9}$$

$$E(P) = \frac{|TP|}{|P|} \log \frac{|TP|}{|P|} - \frac{|FN|}{|P|} \log \frac{|FN|}{|P|}, \tag{10}$$

$$E(Q) = \frac{|FP|}{|Q|} \log \frac{|FP|}{|Q|} - \frac{|TN|}{|Q|} \log \frac{|TN|}{|Q|}, \tag{11}$$

$$P = TP + FN, \tag{12}$$

$$Q = FP + TN, \tag{13}$$

where $E(T)$ measures the information entropy of the original training set T, and $E(T_{\langle p_k^x, p_j^y \rangle})$ measures the information entropy after using $\langle p_k^x, p_j^y \rangle$ to classify instance pairs in T. If $IG(p_k^x, p_j^y)$ is higher than a threshold, $\langle p_k^x, p_j^y \rangle$ is considered as a *discriminative property pair*. We set the threshold 0.2 in our tool. njuLink gets a set of discriminative property pairs after checking all property pairs in candidate property pair set.

Find Coreferent Instances. The key ideas to find coreferent instances are from two aspects: (1) Get detailed similarity w.r.t. an instance pair; and (2) Find the most coreferent instance pair, e.g., for an instance i and an instance set $instSet$, we assume that every instance in $instSet$ seems to be coreferent with i. To find the real coreferent instance pair, first, we use every instance in $instSet$ to form an instance pair with i, and then, we compare the detailed similarity of each instance pair formed and only record the instance pair with highest similarity. It guarantees 1:1 equivalence relationship between two datasets.

Let $DiscrPropSet(D^x, D^y)$ denote the set of discriminative property pairs. First, for every instance in D^x, njuLink combines it with every instance in D^y

to generate many instance pairs, and for every generated instance pair $\langle i_m^x, i_n^y \rangle$, njuLink finds the set of similar discriminative property pair for it, which is denoted by $SimDiscrPropSet(i_m^x, i_n^y)$:

$$\begin{aligned} SimDiscrPropSet(i_m^x, i_n^y) = &\{\langle p^x, p^y \rangle \mid \langle p^x, p^y \rangle \in DiscrPropSet(D^x, D^y) \\ &\& \; p^x \in Prop(i_m^x) \; \& \; p^y \in Prop(i_n^y) \\ &\& \; MaxPropSim(p^x, p^y \mid i_m^x, i_n^y) > 0.65\}. \end{aligned} \tag{14}$$

where $Prop(i_m^x)$ and $Prop(i_n^x)$ are the sets of properties of i_m^x and i_n^y, respectively. Secondly, njuLink calculates seven features below to represent the similarity of the pair:

1) The size of $SimDiscrPropSet(i_m^x, i_n^y)$.
2) The sum of information gain of each similar discriminative property pair $IGSum(i_m^x, i_n^y)$:

$$IGSum(i_m^x, i_n^y) = \sum_{\langle p_k^x, p_j^y \rangle \in SimDiscrPropSet(i_m^x, i_n^y)} IG(p_k^x, p_j^y), \tag{15}$$

3) The sum of detailed information gain of each similar discriminative property pair $DIGSum(i_m^x, i_n^y)$:

$$DIGSum(i_m^x, i_n^y) = \sum_{\substack{\langle p_k^x, p_j^y \rangle \\ \in SimDiscrPropSet(i_m^x, i_n^y)}} DIG(p_k^x, p_j^y \mid i_m^x, i_n^y), \tag{16}$$

$$\begin{aligned} DIG(p_k^x, p_j^y \mid i_m^x, i_n^y) = &(|MatVal(p_k^x, p_j^y \mid i_m^x, i_n^y)| \\ &- |UnmatVal(p_k^x, p_j^y \mid i_m^x, i_n^y)|) * IG(p_k^x, p_j^y), \end{aligned} \tag{17}$$

where $DIG(p_k^x, p_j^y \mid i_m^x, i_n^y)$ denotes the detailed information gain of a similar discriminative property pair w.r.t. $\langle i_m^x, i_n^y \rangle$.

4) The sum of detailed similarity sum of each similar discriminative property $DSimSum(i_m^x, i_n^y)$:

$$DSimSum(i_m^x, i_n^y) = \sum_{\substack{\langle p_k^x, p_j^y \rangle \\ \in SimDiscrPropSet(i_m^x, i_n^y)}} DSim(p_k^x, p_j^y \mid i_m^x, i_n^y), \tag{18}$$

$$\begin{aligned} DSim(p_k^x, p_j^y \mid i_m^x, i_n^y) = &MatValSimSum(p_k^x, p_j^y \mid i_m^x, i_n^y) \\ &* IG(p_k^x, p_j^y), \end{aligned} \tag{19}$$

where $DSim(p_k^x, p_j^y \mid i_m^x, i_n^y)$ denotes the detailed similarity sum of a similar discriminative property w.r.t. $\langle i_m^x, i_n^y \rangle$.

5) The number of similar common property pairs.
6) The sum of maximum similarity of each similar common property pair w.r.t. $\langle i_m^x, i_n^y \rangle$.

7) The number of property pairs that two properties of each one have the same local names. We make every property in $Prop(i_m^x)$ and every property in $Prop(i_n^y)$ a property pair and check them all.

Besides discriminative property pairs, we also use three features from common property pairs because we find discriminative property pairs are not enough to separate the most coreferent instance pairs from those that seem to be coreferent. A common property pair should meet two requirements: this property pair is not a discriminative property pair and two properties of it have the same local names.

Thirdly, njuLink sorts the instance pairs generated in descending order according to these seven scores of each one. The importance of these seven features is 1) > 2) > 3) > 4) > 5) > 6) > 7). Finally, njuLink selects instances in sorted instance pairs set from top to bottom, meanwhile, when we pick up instance pairs from top to bottom, if two instances of an instance pair are both the first time to be checked, we record it, otherwise, drop it. It guarantees the 1:1 equivalence relationship between two datasets D^x and D^y.

1.3 Link to the system and parameters file

You can find the source code and the jar tested by SEALS client successfully on GitHub: `https://github.com/nju-websoft/njuLink`.

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

The alignment files for DOREMUS task should be available at the official website: `http://islab.di.unimi.it/content/im_oaei/2017/`.

2 Results for DOREMUS

There are two sub-tasks under DOREMUS, namely HT and FPT. HT aims to obtain 1:1 equivalence relationship between instances whose data have different types of heterogeneities, while FPT aims to get the same relationship as that of HT between instances with high similarity.

njuLink succeeds in finding property pairs with high discriminability, which are shown in Table 1. The results of evaluation are shown in Table 2 and Table 3.

3 Discussions about improvements

How to apply different approaches according to different datasets automatically? During the development of njuLink, we adjust the way to find coreferent instances according to the requirements of DOREMUS. But the adjusted approach is not applicable for all tasks. So, finding a way to decide appropriate approaches automatically is necessary.

Table 1. Discriminative property pairs on the DOREMUS task

	Properties in dataset 1	Properties in dataset 2
HT	U70_has_title	U70_has_title
	U70_has_title	label
	label	label
	label	U70_has_title
	U16_has_catalogue_statement	U16_has_catalogue_statement
FPT	U70_has_title	U70_has_title
	U70_has_title	label
	label	label
	label	U70_has_title

Table 2. Results for HT

	Precision	Recall	F1-score
AML	0.851	0.479	0.613
I-Match	0.680	0.071	0.129
Legato	0.930	0.920	0.930
LogMap	0.406	0.882	0.556
njuLink	0.966	0.945	0.955

Table 3. Results for FPT

	Precision	Recall	F1-score
AML	0.914	0.427	0.582
I-Match	1.000	0.053	0.101
Legato	1.000	0.980	0.990
LogMap	0.119	0.880	0.210
njuLink	0.959	0.933	0.946

4 Conclusion

njuLink is dedicated to finding coreferent instances by utilizing discriminative property pairs. The Instance Matching track of this year show many new things to us. This helps us find the weaknesses of njuLink and makes our original ideas better. Technical problems happened during the development also forced us to pay more attention to the way of realizing our tool. In the future, we will continue following the trends of instance matching with interests and try to solve issues on which we have not achieved good performance.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (No. 61370019). During our development, we received much support from organizers and volunteers of OAEI, we would like to thank them for their help.

References

1. Hu, W., Jia, C.: A bootstrapping approach to entity linkage on the semantic web. Journal of Web Semantics 34, 1–12 (2015)
2. Hu, W., Yang, R., Qu, Y.: Automatically generating data linkages using class-based discriminative properties. Data & Knowledge Engineering 91, 34–51 (2014)
3. Stoilos, G., Stamou, G., Kollias, S.: A string metric for ontology alignment. In: ISWC 2005. pp. 624–637. Springer (2005)

ONTMAT: Results for OAEI 2017

Saida Gherbi[1]and M[ed]Tarek Khadir[2]

[1]LabGed, ESTI, Annaba 23000, Algeria
Saida_gharbi23@yahoo.fr
[2]LabGed, University Badji Mokhtar Annaba, 23000, Algeria
Khadir@labged.net

Abstract: This paper describes ONTMAT an ontology matching system, and presents the results obtained for the Ontology Alignment Evaluation Initiative (OAEI) 2017. ONTMAT is an ontology matching process, which compares the instances of ontologies to align in order to deduce the relations between their concepts. Then, based on hierarchical and binary relations between the concepts inside the ontologies it performs entities matching.

Keywords: Ontology, Alignment, OWL.

1 Presentation of the system

ONTMAT (ONTology MATching) is an ontology alignment tool, aiming to align OWL entities (classes, object properties i.e. binary relations), participating for the first time in OAEI (Conference track).

1.1 State, purpose, general statement

ONTMAT uses a terminological methods based on WordNet dictionary [2],which is exploited as background knowledge to provide a set of the relations between individuals names of the ontologies source ($O1$) and target($O2$). Then, if the name does not exist in WordNet the approach handles the n-gram measure instead of the dictionary. Moreover, from this set of individual relations we will deduce the equivalence or subsumption relation among their concepts. The equivalent concepts are recorded in an alignment matrix (A*M*), and the concepts related by subsumptions relations are registered within a temporary alignment matrix (*TAM*) [4].

Furthermore, the *TAM* elements and the concepts neighbors of A*M* are compared by using the inference roles with the terminological techniques cited previously and the retained alignment will be added to *AM*. The concepts neighbors are those related by hierarchical or binary relations with AM concepts. Here, we first align the neighboring concepts because they have more chance to be similar [1], after we will align the other concepts by using the same technics. Next, inference technics are applied on *AM* to align the binary relations.

1.2 Approach description

In our proposition we suppose that Wordnet is hierarchically organized as W(S,≤,A_g,g), where S is a set of synsets {s_1,s_2,...,s_i}(i is a positive integer), and an annotate function A_g associates the gloss g to each synset. Furthermore, the relations ≤ between concepts s_1, s_2 may be presented in the following logical relations [4] as:

1) $s_1 \subseteq s_2$;means thats_1 is a hyponym or meronym of s_2;
2) $s_1 \supseteq s_2$; express that s_1 is a hypernym or holonym of s_2;
3) $s_1 \equiv s_2$; signified that s_1 and s_2 belong to the same synset are similar[1];
4) $s_1 \perp s_2$ when s_1 and s_2 are the siblings in the part of hierarchy they are connected by a relation of antonymy.

The entities aligned can be related by one of the hierarchical relation presented in the set $HR = \{\equiv, \subseteq, \supseteq\}$where (≡: equivalence; ⊆:subclass), fuzzy relation symbolized by "&", or binary relation. Further, the binary ontologies relations (O_1, O_2)are also aligned by an element of the set *HR*. The algorithm will explain in the following items:

1. In level 1 we compare the instances names (I_{O1},I_{O2}) of ontologies($O1, O2$)to deduce the relations among their concepts. To do this, WordNet is exploited because we cannot assume with certainty that two entities are dissimilar if they have different names (synonyms), or they are equivalent if they have the same name(homonyms). If the name does not existed in WordNet we will measure the similarity among names by the n-gram measure. Then the equivalent instances will construct the instances matrix *IM*. The concepts (C_1,C_2) of ($O1, O2$) that have the same sets of instances in *IM* are considered as equivalent concepts as proven in [4], and ($C_1, C_2, \equiv$) can be added to *AM*. Although, if the instances set of C_1 are included in the instances set of C_2then ($C_1, C_2, \subseteq$) will be inserted in *TAM*.
2. The level 2 starts by applying terminological techniques on the concepts names. Next the results obtained will be combined with inferences methods illustrated in [4] to be inserted in *AM*: (C_1, C_2, Rel_1) of *TAM* confirmed or modified, where $Rel_1 \in \{\equiv, \subseteq, \supseteq, \&\}$.
3. The concepts neighbors sorted by hierarchical relations of the *AM* elements (C_1, C_2, Rel_1), are (C_1', C_2')linked to (C_1, C_2)respectively by an element of the set *HR*. The neighbors(C_1', C_2')joined by "≡" with(C_1, C_2)of AM in ($O1, O2$), will be aligned by using the inferences techniques applied on the background knowledge [3]. The background knowledge is the ontology source when the neighbors belong to O1, and ontology target if we match the neighbors existing in O2. The other neighbors will be matched using the terminological methods.
4. The fourth level exploits the description logic roles proven in [4] to match the concepts (C_1', C_2') associated to (C_1, C_2) by binary relations (B_1, B_2) in ($O1, O2$), as following:

- If $(C_1, C_2, \equiv) and\ (C_2, C_2', B_1)$ then (C_1, C_2', B_1) will be inserted to A_{BRS}M (Alignment Binary Relation Source Matrix)
- If $(C_1, C_2, \equiv) and\ (C_1, C_1', B_2)$ then (C_2, C_1', B_2) will be added to A_{BRT}M (Alignment Binary Relation Target Matrix)

Thus, binary relations can be aligned because we have: $B_1 \mathrm{h_R} B_2$ Iff dom(B_1) $\mathrm{h_R}$ dom(B_1) and ran(B_1) $\mathrm{h_R}$ ran(B_1); where $h_R \in HR$[4], for instance;
If $(C_1, C_2, \subseteq) and\ (C_1, C_1', B_1)$and $(C_1', C_2', \subseteq) and\ (C_2, C_2', B_1)$ then $(B_1, B_2, \subseteq)$ will be added to A_{BR} (Alignment Binary Relation Matrix).

5. Finally, the concepts not yet aligned, will be matched via the terminological methods.

1.3 Adaptations made for the evaluation

We have adapted the format of the alignment result to the reference alignments restricted to name classes, using the "=" sign for equivalence relation with confidence of 1. Although our system provides other relations as subsumption, and binary relations without measure, as well as the alignment of binary relation by the *HR*.

2 Results

In this version we wish to test the techniques used by ONTMAT, such as, the inferences mechanisms applied onWordNet and the ontologies source and target, and the deduction of the matching among entities based on instances. The most appropriate track to do these tests is the conference track.
Conference track comprises 16 ontologies from the domain of conference organization. Most ontologies of this track were equipped with OWL DL axioms; which is useful to test our inferences approach. Table 1shows the evaluation result obtained by running ONTMAT under the SEALS client with the command:
java -jar F:/temp/seals-omt-client.jar F:/temp/ONTMAT -t

This command tests two predefined ontologies from the Conference. From Table 1 we can write that ONTMAT perform well because these ontologies are the same structure.

Table 1. Results for two predefined ontologies

Precision	Recall	F-Measure
1.0	0.455	0.625

The results obtained by the global test as illustrated in Table 2, are not well as the results of the precedent table in term of precision and F-measure. Although, the global recall is *0.434*.

Table 2. Results for conference track

Test Case ID	Precision	Recall	F-measure
cmt-conference	0.6	0.2	0.3
cmt-confof	0.4	0.25	0.308
cmt-edas	0.444	0.615	0.516

cmt-ekaw	0.217	0.455	0.294
cmt-iasted	0.143	1.0	0.25
cmt-sigkdd	0.176	0.5	0.26
conference-confof	0.052	0.467	0.094
conference-edas	0.052	0.412	0.092
conference-ekaw	0.059	0.32	0.1
conference-iasted	0.03	0.286	0.054
conference-sigkdd	0.059	0.533	0.106
confof-edas	0.04	0.421	0.073
confof-ekaw	0.04	0.4	0.073
confof-iasted	0.02	0.444	0.038
confof-sigkdd	0.02	0.571	0.039
edas-ekaw	0.016	0.217	0.03
edas-sigkdd	0.022	0.467	0.042
ekaw-iasted	0.015	0.6	0.029
ekaw-sigkdd	0.017	0.636	0.033
iasted-sigkdd	0.02	0.733	0.039
Global	0.034	0.434	0.063

2.1 Discussions on the way to improve the proposed system

To improve our application, we will also align the properties of ontologies ($O1, O2$). Then, adapt it to read all files type, and integrate the translator to test our tool under other tracks as: Instance Matching, MultiFarm.

2.2 Comments on the OAEI test cases

The application seals-omt-client from seal, only test files where the alignment relation between concepts is itself the equivalence relation. However ONTOMAT, offers other possibilities in terms alignment relations between entities such as; & : Fuzzy and binary relations. We hope that OAEI takes into consideration those types of relations in the reference alignment file.

3 Conclusion and future work

We have briefly described the mechanisms exploited by our proposition ONTMAT, and presented the results obtained under the conference track of OAEI 2017.
This is our first participation in OAEI, the results are not satisfying, and the system presents some limitations. In the future, we will make great efforts to improve ONTMAT results, and participate in more tracks.

References

1. Euzenat, J., Shvaiko. P. “Ontology Matching” , pages 37-39; 73-87, 92-93. Springer-Verlag Berlin Heidelberg, 2007.
2. Fellbaum, C. :WordNet: An Electronic Lexical Database, MIT Press, Cambridge, MA (1998).
3. GherbiS., BelleiliH.andKhadirM., 2013. BRMAP : Un outil d’Alignement des ontologies. 7ème édition de la conférence maghrébine sur les avancés des systèmes décisionnels.
4. GherbiS., 535 M. T. Khadir, Inferred ontology concepts alignment using and an external dictionary, Procedia Computer Science 83 (2016) 648- 652, the 7th International Conference on Ambient Systems, Networks and Technologies (ANT 2016) / The 6th International Conference on Sustainable Energy Information Technology (SEIT-2016) / Aliated Workshops.doi:http://dx.doi.org/10.1016/j.procs.2016.04.145.URL http://www.sciencedirect.com/science/article/pii/ S1877050916301752.
5. Giunchiglia, F., Shvaiko, P., Yatskevich, M., 2004. SMatch an algorithm and an implementation of semantic matching. In Proc. 1st EuropeanSemanticWeb Symposium (ESWS), volume 3053 of Lecture notes in computer science, pages 61–75, Hersounisous (GR), 10-12 May 2004.

POMap results for OAEI 2017

Amir Laadhar[1], Faiza Ghozzi[2], Imen Megdiche[1], Franck Ravat[1], Olivier Teste[1], and Faiez Gargouri[2]

[1] Paul Sabatier University, 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 an effective strategy to find the correspondences among different ontologies in a scalable and a heterogeneous semantic web. In order to find these correspondences, a matching system should be built aiming to ensure the interoperability between ontologies. POMap (Pairwise Ontology Mapping) is an automated ontology matching system dealing with the three main types of heterogeneity: syntactic, semantic and structural. During our first participation in the OAEI campaign, POMap succeeded to be one of the top three performing systems in the Anatomy track. In the remaining of this paper, we briefly introduce POMap and discuss its OAEI 2017 results according to four tracks: Anatomy, Conference, Large Biomedical Ontologies, Disease and Phenotype.

Keywords: Semantic web, ontology matching, semantic matching, syntactic matching, structural matching

1 Presentation of the system

1.1 State, purpose, general statement

An ontology can model a particular domain as well as the semantic relationships between its entities in order to ensure its reuse by different stakeholders. Several ontologies describing the similar domain can be generated and used by various parties defined by different terminologies. Despite the standardization of the ontology representation, the heterogeneity problem emerges. Therefore, it is important to overcome this heterogeneity to ensure the reusability of various ontologies. Indeed, many researchers has been proposing and developing many automated ontology matching systems. Ontology matching is the process of finding a set of correspondences between the entities of two or more ontologies representing a similar domain. Therefore, these systems are using a variety of strategies relying on the combination of several techniques such as: Syntactic, semantic and structural based strategies. As depicted in figure 1, POMap is pursuing a sequential composition during the mentioned three matching techniques. POMap is exploring all these three techniques in order to ensure a high quality

matching. Only dealing with the anatomy track, we employ a semantic matcher. Then, for all the other OAEI tracks, we used a syntactic matcher, which follows an all-against-all strategy. Next, our structural matcher takes as an input the generated mappings from the semantic matcher and the syntactic matcher in order to find new correspondences. The adopted sequential composition aims to prune the search space used by the structural matcher. This structural matcher is composed of two structural sub-matchers: siblings and subclasses. A broader explanation of POMap could be found in [1]. In the next subsection, we will briefly describe each component of our system as well as the used techniques.

1.2 Specific techniques used

The POMap workflow for our first participation on the OAEI comprises three main steps, as flagged by the figure 1: Ontology indexing and loading, ontology matching and output alignment generation.

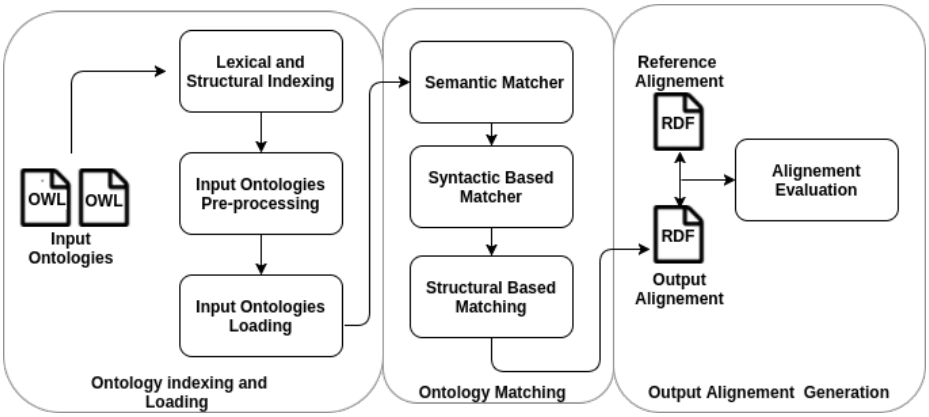


Fig. 1. The architecture of POMap.

Step 1: Ontlogy indexing and loading

The initial step of POMap is the extraction of all the annotations within the two input ontologies. In terms of lexical indexing, POMap builts a multimap data structure that contains the triplet: the set of entities, their annotations as well as the property type of each annotation. For the structural indexing, all relationships between the extracted entities are stored in a multimap data structure. Every record of this multimap contains two entities and the relationship property between them. After accomplishing the lexical and the structural indexing, we perform several preprocessing strategies, such as: the removal of non-alphanumeric characters, the removal of stopwords, the stemming process and the lowercasing.

Step 2: Ontology matching

Step 2.1: The semantic Matcher

The first step in the matching process is performing the semantic matcher. We argue this choice by the high precision of the adopted semantic matcher. Therefore, we will be based on it to enrich the resulted mappings by new ones through the use of syntactic and structural strategies. During this first participation in the OAEI campaign, we adopted the semantic matching only for the Anatomy track. We plan to expand the use of this matcher in our future participation. In order to ensure the semantic matching, we employed Uberon [3] as an external biomedical knowledge source for the alignment of the Anatomy track. Uberon is an integrated cross-species ontology covering anatomical structures and includes relationships to taxon-specific anatomical ontologies. Indeed, we explored the property "hasDbXref", which is mentioned in almost every class of Uberon. This property references the classes' URI of some external ontologies such as the human and mouse of the Anatomy track. Consequently, we align every two entities of the Anatomy track in case if they are both referenced in a single class of Uberon.

Step 2.2: The syntactic Matcher

After performing the semantic matching process, we are able now to apply the syntactic matcher. This syntactic matcher computes the similarity score between every two names of the two input ontologies using a string similarity measure. The variety of the existing state of the art similarity measure arises the problem of choosing the right one associated with its optimal threshold. Therefore, we tested the available syntactic similarity measure (https://goo.gl/1kUgkH) while variating the associated threshold value. Hence, we selected ISUB combined with a threshold of 0.9. Only the couple of entities having a similarity score above 0.9 are considered as new mappings candidates. As we are performing a pairwise (1:1) matching process, for every single entity from the first ontology, we select only one entity with the maximum similarity score. In case of two candidate mappings have the exactly same similarity score, we consider randomly one of them as the final alignment.

Step 2.3: The structural Matcher

For the set of available correspondences derived from the semantic and the syntactic matcher, we are able to enrich them by a set of new correspondences through the use of the structural matching. This structural matcher is composed of two sub-matchers based on siblings and subclasses.

Step 2.3.1: The structural Matcher based on siblings

For the structural matcher based on siblings, we follow the intuition of: if two entities match, then their sibling should somehow similar [2]. Therefore, if two entities are aligned using the syntactic matcher, we compute the similarity score between their siblings. Then, following an alignment multiplicity of 1:1, we match the siblings having a similarity score between ISUB 0.9 (syntactic threshold) and ISUB 0.8. The resulted mappings from the structural matcher based on siblings are added to the already discovered correspondences by the two earlier matchers.

Step 2.3.2: The structural Matcher based on subclasses

Concerning the structural matcher based on subclasses, we pursue the intuition that if two classes are similar, then their subclasses should be similar [2]. This intuition should be straightforward applied if two classes are having a very small number of subclasses. Nonetheless, this will be complicated in case of there are many descendants. Therefore, as a first step, we remove all the common tokens between an already aligned entity and its descendants. We argue that there is a syntactic inheritance between an entity an their descendants. Therefore, the removal of these similar tokens, will permits to better capture the similarity between two entities. Then, we compute the similarity score among all the descendants of two already aligned entities while applying the similarity measure of Monge Elkan 0.85 [4]. Unlike ISUB, we argue the use of Monge Elkan due to its particularity in capturing the dissimilarity between two textual sequences containing numerical values. However, this similarity measure is not recommended for a heavy matching process, due to its time consuming.

Step 3: Output alignment generation

As a final step, we generate an RDF file, which contains the alignment based on the resulted mappings resulted by all the employed matchers.

1.3 Link to the system and parameters file

The SEALS wrapped version of POMap for the OAEI 2017 is available at: https://goo.gl/mZ4PzR

1.4 Link to the set of provided alignments

The resulted alignments by POMap as well as the results for each track during our participation in OAEI 2017 are available at: https://goo.gl/mZ4PzR.

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 93.3%, which is very close to the top results. We argue the importance of the obtained results by the effectivenesses of the overall employed matchers, the use of all the names of the input ontologies and applying an efficient preprocessing process. 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.

Table 1. POMap results in the anatomy track compared to the OAEI 2017 systems.

System	Precision	Recall	F-Measure	Runtime
AML	0.95	0.936	.943	47
YAM-BIO	0.948	0.922	0.935	70
POMap	0.94	0.925	0.933	808
LogMapBio	0.889	0.899	0.894	820
XMap	0.926	.836	.893	37

2.2 Conference

The purpose of the conference track is to find the correspondences within a collection of ontologies describing the domain of organizing conferences. Matching systems are evaluated according to the combination of three reference alignments along with three evaluation modalities (M1,M2 and M3). These evaluation modularities are containing respectively: only classes, properties as well as classes and properties. Since we did not focus on the matching of properties, the table 2 draws the obtained results by POMap results only for the first modularity and partially for the third modularity. Therefore, we plan for our next participation in the OAEI to include the property matching in order to make a more comprehensive evaluation of this track.

2.3 Large biomedical ontologies

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

Table 2. POMap results for the conference track

	Precision	Recall	F1-Measure
Ra1-M1	0.88	0.47	.61
Ra1-M3	0.73	0.4	0.52
Ra2-M1	0.83	0.43	0.57
Ra2-M3	0.67	.37	.48
Ra2-M1	0.889	0.899	0.894
Ra2-M3	0.69	0.38	0.49

fragments) and FMA-SNOMED (small fragments) with an F-Measure respectively of 86.1% and 41.6%. For the other tasks of the large biomedical track, POMap exceeded the defined timeout. As a future work, we are planning to cope with the matching process of the larger ontologies in a shorter time.

2.4 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 i7-4600U CPU @ 2.10GHz x 4 coupled with 15Gb RAM. Due to the timeout limit, POMap succeeded to complete tow tasks (HP-MP and DOID-ORDO) out the four tasks of this track. POMap produced 2024 mappings in the HP-MP task associated with 402 unique mappings. Among twelve matching systems, POMap achieved the fifth highest F-measure according to the 2-vote silver standard, with an F-Measure of 73.2%. In the DOID-ORDO task, POMap generated 3222 mappings with 666 unique ones. According to the 2-vote silver standard, it scored an F-Measure of 80.5%.

3 Conclusion

The first version of POMap ontology matching system as well as its obtained results in the OAEI campaign were presented in this paper. We proposed three matchers: semantic, syntactic and structural. We performed the structural matching without any propagation syntactic similarity score or computation of a structural similarity score. We are guided only by the syntactic treatment of both subclasses and siblings. The obtained results 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 match larger ontologies in the given runtime threshold. Consequently, we are planning to optimize our matching system for larger biomedical tasks while taking into consideration the automatic tuning of the matching configuration.

References

1. A. Laadhar, F. Ghozzi, I. Megdiche, F. Ravat, O. Teste, F. Gargouri POMap: An Effective Pairwise Ontology Matching System 9th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (KEOD'17), Funchal (Madeira, Portugal) 2017
2. Shvaiko, P., Euzenat, J. (2013). Ontology matching: state of the art and future challenges. IEEE Transactions on knowledge and data engineering, 25(1),
3. Mungall, Christopher J., et al. "Uberon, an integrative multi-species anatomy ontology." Genome biology 13.1 (2012): R5.
4. Monge, Alvaro E., and Charles Elkan. "The Field Matching Problem: Algorithms and Applications." KDD. 1996.

RADON results for OAEI 2017

Kevin Dreßler[2], Mohamed Ahmed Sherif[1], and Axel-Cyrille Ngonga Ngomo[1]

[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. Datasets containing billions of geospatial resources are increasingly being represented according to the Linked Data principles. RADON is an efficient solution for the discovery of topological relations between such geospatial resources according to the DE9-IM standard. RADON uses a sparse space tiling index in combination with minimum bounding boxes to reduce the computation time of topological relations. In this paper, we present the participation of RADON in the OAEI 2017 campaign. The OAEI results show that RADON outperforms the other state of the art significantly in most of the cases.

1 Presentation of the system

RADON is a time-efficient link discovery algorithm for topological relations between geospatial resources, implemented within LIMES [3].

Given two sets of RDF resources S and T and a relation R, the goal of link discovery is to find the *mapping* $M = \{(s,t) \in S \times T : R(s,t)\}$. RADON enables the time-efficient discovery of all topological relations that can be defined in terms of the DE-9IM standard [1]. In order to achieve time-efficiency, two optimization techniques are utilized: *optimized sparse space tiling* on the dataset level and *Minimum Bounding Box (MBB)-based filtering* on the resource level.

In the following, we introduce the basic concepts needed to understand RADON before we outline the aforementioned optimization techniques. More detailed explanations can be found in [5].

The Minimum Bounding Box (MBB) of a geometry g in n dimensions is the rectangular box with the smallest measure (area, volume, or hypervolume in higher dimensions) within which all points of g lie. Another term for MBB is *envelope*.

Space tiling is a technique for indexing spatial data, where n-dimensional affine spaces are split into any number of hyperrectangles with edge lengths ℓ_i and granularity factors $\Delta_i = (\ell_i)^{-1}$ where $i \in \{1, \ldots, n\}$. These hyperrectangles can then be addressed using vectors from $\mathbb{N}^n$, which allows for various optimizations.

1.1 Optimized Sparse Space Tiling

The goal of the *optimized sparse space tiling* is to generate an index I for mapping all geometries $s \in S, t \in T$ to sets of hyperrectangles. For the sake of clarity, the following description focuses on the two-dimensional case. As a first step, we use a heuristic to get good granularity factors for both latitude and longitude dimensions ($\Delta_\varphi, \Delta_\lambda$). Then, we apply space tiling, in which we map a geometry g to the set of hyperrectangles over which its MBB spans. To implement this idea, we insert a reference to g into all those hyperrectangles, that are realized as entries of a `HashMap`. To optimize (i.e. sparsify) the generated index, we start by computing estimated total hypervolumes (ETH) of the datasets S and T. We first index the dataset with the smaller ETH for each resource of the other dataset. We then add only to I the subset of resources from the second dataset which shares the same hyperrectangles from the first dataset resources contained in I. Using this technique together with the `HashMap` implementation of the hyperrectangle index significantly reduces the size of the generated data structure and consequently also the time to traverse it.

1.2 MBB-based Filtering

After the *optimized sparse space tiling* step described above, we traverse the generated index, visiting one hyperrectangle at a time. As a consequence of our approach, each generated hyperrectangle contains references to at least one geometry from each dataset. For each pair (s, t) of geometries, where $s \in S$ and $t \in T$, we then employ a filtering step before actually triggering the potentially expensive (in cases of large geometries) computation that checks if the given relation holds. Let $\square(g)$ denote the MBB of geometry g. The filtering step leverages the fact that $\neg r(\square(s), \square(t)) \Rightarrow \neg r(s, t)$ holds for every relation r, where one geometry has no interior or boundary points in the exterior of the other geometry, i.e. $s \subseteq t$ or $t \subseteq s$. For these relations, we can return false and skip further computations, iff the geometries MBB's do not satisfy the relation.

2 Adaptations made for the evaluation

No specific adaptations were made to the original RADON algorithm [5], we only provide a Java `SystemAdapter` according to the campaign guidelines[3]. The final RADON Java `SystemAdapter` source code is available online in the project website[4].

3 Evaluation Results

RADON 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: *Disjoint, Touches, Contains/Within, Covers/CoveredBy,*

[3] `https://goo.gl/cWmZ5P`
[4] `https://goo.gl/awkvvo`

Intersects, Crosses, Overlaps. The geospatial resources traces were represented in Well-known text (WKT) format as `LineStrings` .

Given two sets of `LineString` geometries S and T and a DE-9IM topological relation R, the participants were assigned the task of retrieving the mapping $M = \{(s,t) \in S \times T : R(s,t)\}$. 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 2K instances.

The other participants to this task in addition to Radon were AgreementMakerLight (AML), OntoIdea, and Silk. The systems were judged on the basis of precision, recall, F-Measure and run time. The final results are shown in Table 1 and Figures 1 and 2. Note that we are only presenting the time performance and not precision, recall and F-Measure, as all were equal to 1.0 except OntoIdea Touches and Overlaps which is equal to 0.99.

From these results we can see that, while Radon performs in the middle field of the the sandbox dataset, Radon outperforms the other participants on most relations for the sandbox dataset. Notably, the optimization described in Section 1.2 speeds up the relations *Equals, Contains, Within, Covers* and *CoveredBy* significantly in comparison to the remaining relations. The differences in performance between *Touches, Intersects*, where AML outperforms Radon, and *Overlaps* cannot be explained from an implementation point of view, as these three 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 *Overlaps*. Therefore, the benchmarks on these relations are the hardest in this task.

4 Conclusion

We priefly presented Radon, an approach for rapid discovery of topological relations among geo-spatial resources. To achieve a high scalability, Radon combines space tiling, minimum bounding box approximation and a sparse index. The presented evaluation during the OAEI 2017 showed that, in addition to being complete and correct (i.e. achieving an F-Measure of 1.0), Radon also outperforms the other participating systems in most of the cases. In future work, we aim to apply the particle-swarm-optimization load balancing approaches [6]. To improve the performance of Radon on high resolution datasets, i.e. datasets whose containing geometries consist of a large set of points, we will optimize the computation of relation checks. In order to further reduce the amount of computations, we will consider adaptive granularity factors, i.e. granularity factors as functions of latitude and longitude. In addition, we aim to combine Radon with the machine learning approaches already implemented in Limes such as the Wombat [4] algorithm. Finally, we will consider the discovery of *temporospatial* relations, by integrating the Aegle[2] algorithm with the Radon approach.

Acknowledgments

This work has been supported by the eurostars project SAGE (GA no. E!10882), the H2020 projects SLIPO (GA no. 731581) and HOBBIT (GA no. 688227) as well as the

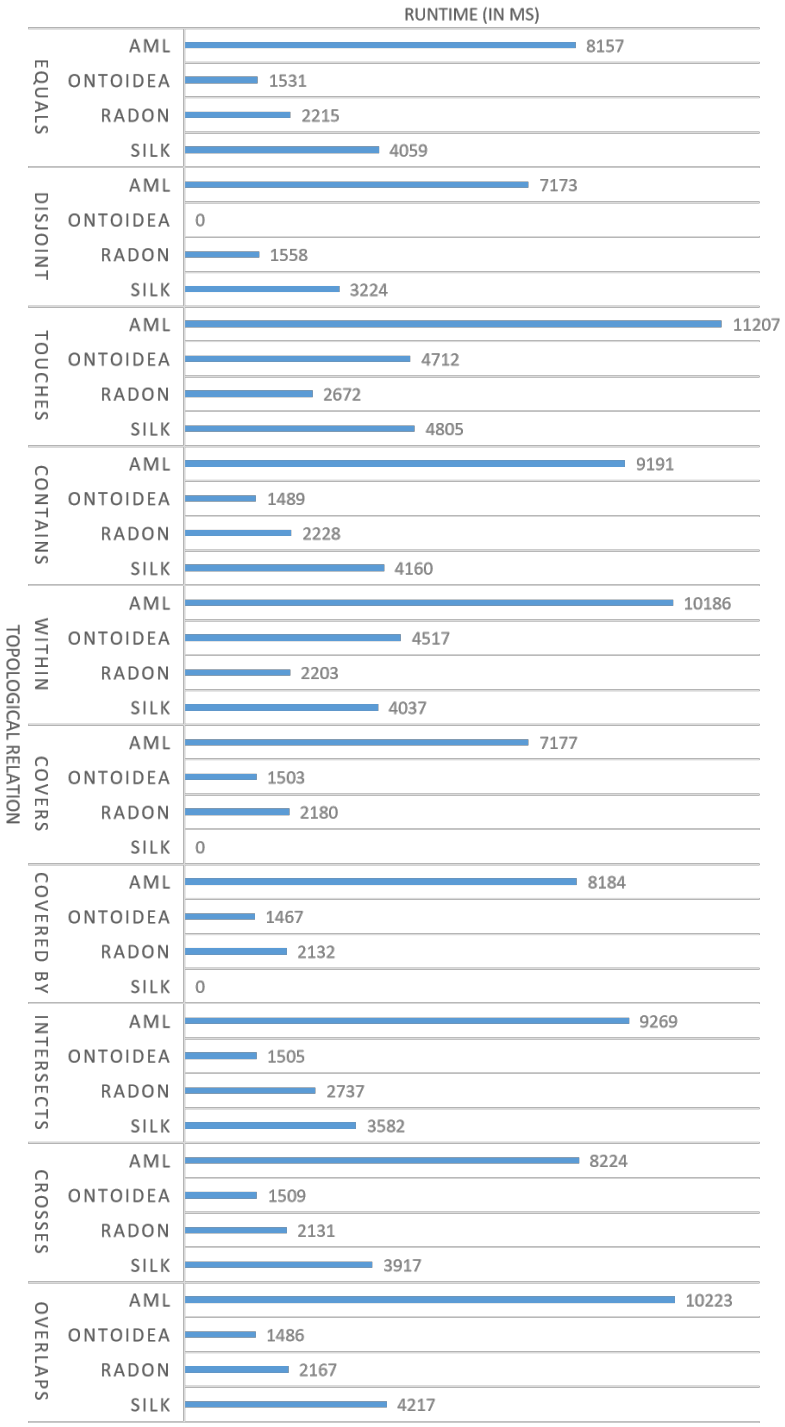


Fig. 1. Runtime comparison for *Sandbox* dataset

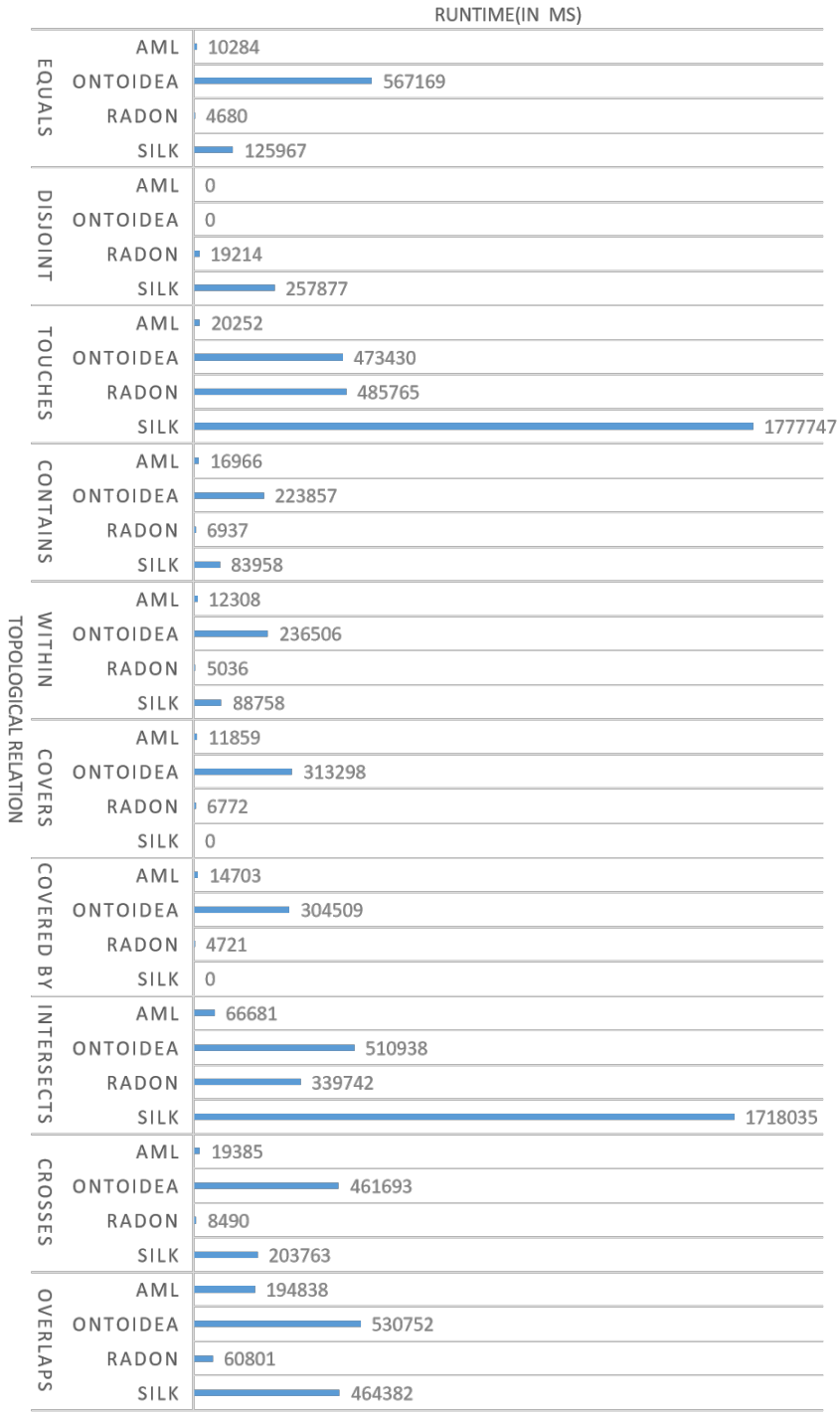


Fig. 2. Runtime comparison for *Mainbox* dataset

Table 1. Hobbit link discovery task evaluation results for all participants. Note that we used — for systems which were not participating in the specified sub-task and × for systems that exceeded the time limit.

Relation	System	Sandbox	Mainbox
Equals	AML	8157	10284
	OntoIdea	1531	567169
	Radon	2215	4680
	Silk	4059	125967
Disjoint	AML	7173	×
	OntoIdea	—	—
	Radon	1558	19214
	Silk	3224	257877
Touches	AML	11207	20252
	OntoIdea	4712	473430
	Radon	2672	485765
	Silk	4805	1777747
Contains	AML	9191	16966
	OntoIdea	1489	223857
	Radon	2228	6937
	Silk	4160	83958
Within	AML	10186	12308
	OntoIdea	4517	236506
	Radon	2203	5036
	Silk	4037	88758
Covers	AML	7177	11859
	OntoIdea	1503	313298
	Radon	2180	6772
	Silk	—	—
CoveredBy	AML	8184	14703
	OntoIdea	1467	304509
	Radon	2132	4721
	Silk	—	—
Intersects	AML	9269	66681
	OntoIdea	1505	510938
	Radon	2737	339742
	Silk	3582	1718035
Crosses	AML	8224	19385
	OntoIdea	1509	461693
	Radon	2131	8490
	Silk	3917	203763
Overlaps	AML	10223	194838
	OntoIdea	1486	530752
	Radon	2167	60801
	Silk	4217	464382

DFG project LinkingLOD (project no. NG 105/3-2) and the BMWI Project GEISER (project no. 01MD16014E).

References

1. E. Clementini, J. Sharma, and M. J. Egenhofer. Modelling topological spatial relations: Strategies for query processing. *Computers & graphics*, 18(6):815–822, 1994.
2. K. Georgala, M. A. Sherif, and A.-C. N. Ngomo. An efficient approach for the generation of allen relations. In *Proceedings of the 22nd European Conference on Artificial Intelligence (ECAI) 2016, The Hague, 29. August - 02. September 2016*, 2016.
3. A.-C. Ngonga Ngomo and S. Auer. Limes - a time-efficient approach for large-scale link discovery on the web of data. In *Proceedings of IJCAI*, 2011.
4. 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.
5. M. A. Sherif, K. Dreßler, P. Smeros, and A.-C. N. Ngomo. Radon-rapid discovery of topological relations. In *AAAI*, pages 175–181, 2017.
6. M. A. Sherif and A.-C. N. Ngomo. An optimization approach for load balancing in parallel link discovery. In *Proceedings of the 11th International Conference on Semantic Systems*, SEMANTICS '15, pages 161–168, New York, NY, USA, 2015. ACM.

SANOM results for OAEI 2017

Majid Mohammadi[a], Amir Atashin[b], Wout Hofman[c], Yao-Hua Tan[a]

[a]Faculty of Technology, Policy and Management, Delft University of Technology, Netherlands.
[b]Department of Computer Engineering, Ferdowsi University of Mashhad, Iran.
[c]TNO research institute, Netherlands.

Abstract

Simulated annealing-based ontology matching [1], or SANOM, is an ontology alignment system which exploits the well-known simulated annealing to find the correspondences. The system considers three different similarity measures, namely string-based, linguistic-based and structural-based measures. A rudimentary version of the proposed method is participated in Ontology Alignment Evaluation Initiative (OAEI) 2017, and the results are report accordingly.

Keywords: SANOM, ontology alignment, OAEI.

1. System Representation

SANOM is an energy-based ontology alignment system which tries to find the most possible alignment through the minimization of a predefined energy function by the well-known simulated annealing method.

To define the energy function for a given alignment, we need to process each existing correspondence. To do so, three different similarity measures is taken into account. For each correspondence in the alignment, the minus sum of all the similarity measures is considered as the energy; therefore, the alignment with minimum energy entails more similar concepts. In the following, the potential similarity measures are reviewed along with the simulated annealing.

1.1. Simulated Annealing

Simulated annealing is a probabilistic approach to estimate the global optimum of problems which cannot be solved by the standard optimization techniques. As the name suggests, this technique simulates the annealing in metallurgy which slowly cool the materials to decrease their defects.

The controlled cooling in the simulated annealing method is implemented as the decrease in the probability of accepting the worse solution. It is fundamental in this algorithm to accept the worse solutions with some probability in order to escape the local optimum.

Let S be the current state and S' be the *successor* (or the neighbor) created based on the current state. Simulated annealing needs a fitness function to estimate the fineness of each state. The transition form the current state to the next is probabilistic: If the successor has a better fitness than the current state, then the transition to the successor will definitely happen (or with the probability of 1.) In other words, the transition to the successor is made if $\Delta Eng = fitness(successor) - fitness(current) > 0$ where ΔEng is the difference between the fitness of two states and *fitness(a)* indicates the fineness of the state a. Otherwise, if the successor is not as good as the current state, e.g. $\Delta Eng < 0$, the transition happens with the probability of $P = e^{\frac{\Delta E}{T}}$ where T is the temperature. It is plain to see that transition to the worse solution is less likely when the temperature is lower. The simulated annealing algorithm starts with higher temperature and gradually decreases the temperature. This means that the probability of transition to the worse solution is way higher at the beginning, and little by little it is less feasible to get the worse solution as the temperature augments.

1.2. Problem Formulation

The ontology alignment is the relation between the concepts of two given ontologies. The relation (or map) could be seen as a bipartite graph, in which each part represents the concepts of one ontology and the edges indicate the similarity among concepts.

Let G be the bipartite graph depicting the relation between the concepts of two given ontologies. Assume that C_1 and C_2 are the concepts of two given ontologies, the nodes of the graph are the concepts of two ontologies, i.e. $V = C_1 + C_2$, and the edges connect each concept from one side of the graph to the other. The weights w of edges are the similarity among the concepts, which can be shown by $w : E \rightarrow R$, where $E \subseteq C_1 \times C_2$.

The cardinality is assumed to be $1 : 1$, meaning that each concept from the first ontology is mapped only with (maximum) one concept from the target. Let $S(e), e \in E$ be the similarity between two arbitrary concepts, the goal could be summarized in the following problem

$$\max \sum_{e \in E} S(e) \tag{1}$$

To gauge the similarity of two given concepts, various measures are considered. The similarity measures are classified into string, linguistic and structural measures. Thus, $S(e)$ can be defined as

$$S(e) = S_s(e) + S_l(e) + S_{st}(e) \tag{2}$$

where $S_s(.)$, $S_l(.)$ and S_{st} are the string, linguistic and structural similarity measures, respectively.

1.3. SANOM

Let the energy function $Eng(.)$ be

$$Eng(E) = \sum_{e \in E} S(e) \tag{3}$$

then the output alignment of the above energy function from the simulated annealing method is the final result of the system.

2. Results

In this section, the results of various tracks in which SANOM has participated are reported.

2.1. The Anatomy track

The Anatomy track is the challenge of matching two different anatomy ontologies from human and mouse. The result of SANOM is compared with other systems via McNemars test. There are two ways to apply McNemars test in which the difference is if we consider the false correspondences or not [2].

Figures 1 shows the directed graph from the outcome of McNemars test over the systems participated in OAEI 2017 while false correspondences are not taken into account. From another angle, Figure 2 shows the same graph but considering the false correspondences. The nodes in the directed graphs are the systems and each directed edge $A \rightarrow B$ indicates that System A is better than System B.

According to these figures, SANOM has outperformed ONTOEMMA, WikiV3 and Alin in both cases while AML, POMap, YAM-Bio and Xmap has a better performance than SANOM. Further, SANOM and KEPLER is quite competitive: If the false correspondences are taken into account KEPLER is better while SANOM is superior if only correct correspondences are taken into account. It means that SANOM has more true and false correspondences than KEPLER.

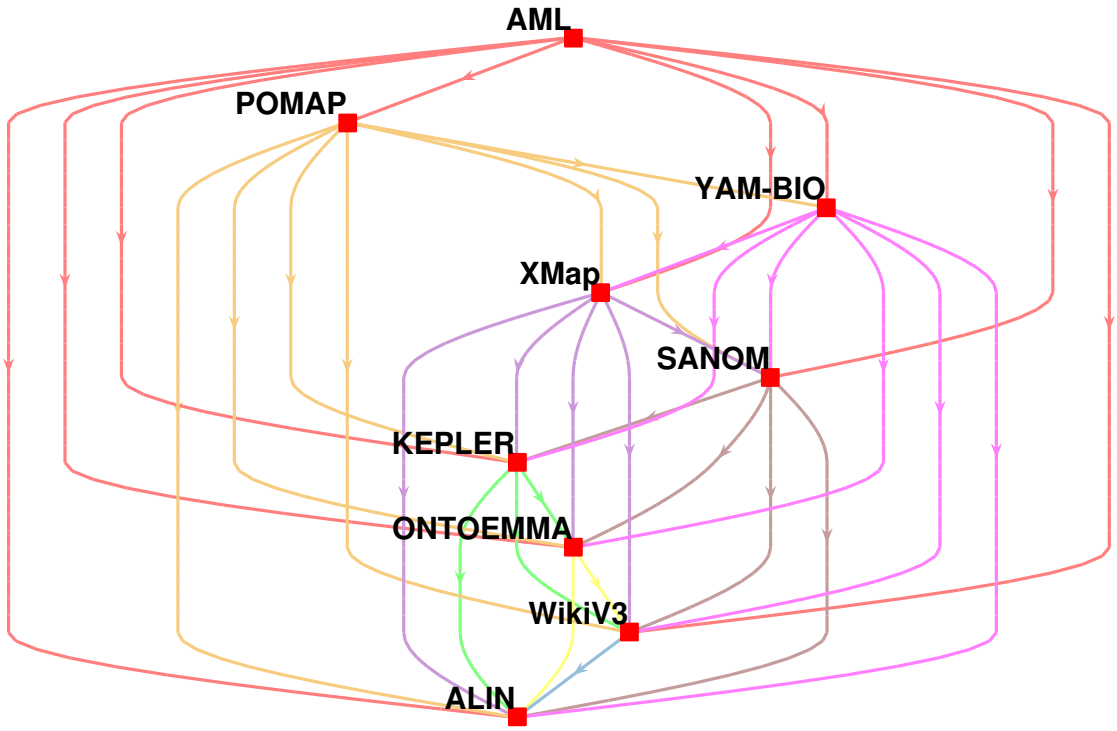


Figure 1: Comparison of alignment systems participated in OAEI 2017 on the anatomy track while the false correspondences are not considered.

Table 1: The average F-measure obtained of the systems over the Conference track

	AML	LogMap	Xmap	KEPLER	LogMapLite	Wiki3	POMap	ALIN	SANOM	ONTMAT
ra1-m1	0.76	0.73	0.73	0.68	0.66	0.66	0.61	0.47	0.43	0.18
ra1-m2	0.58	0.39	0.32	0.21	0.23	0.16	0	0	0	0
ra1-m3	0.74	0.69	0.68	0.59	0.59	0.57	0.52	0.41	0.38	0.11
ra2-m1	0.71	0.67	0.67	0.62	0.6	0.6	0.57	0.44	0.42	0.18
ra2-m2	0.58	0.39	0.35	0.21	0.23	0.16	0	0	0	0
ra2-m3	0.7	0.63	0.63	0.54	0.54	0.52	0.48	0.39	0.37	0.1
rar2-m1	0.71	0.69	0.68	0.63	0.62	0.62	0.58	0.46	0.44	0.18
rar2-m2	0.56	0.4	0.35	0.21	0.23	0.16	0	0	0	0
rar2-m3	0.69	0.66	0.65	0.55	0.56	0.54	0.49	0.41	0.38	0.1

2.2. The Multifarm track

This track includes the alignment between ontologies coming from different languages. SANOM, in the current version, does not use any translator so that it is not able to find good correspondences in this track. However, it has produced some results due to the structural similarity between two ontologies. SANOM is compared with other participants via the Friedman test [3], and the outcome is visualized by the critical difference diagram, as shown in Figure 3. The x-axis in this figure shows the average rank of each system obtained by the Friedman test: The lower the rank, the better the system. The systems with equivalent performance from the statistical point of view are connected to each other by a line.

According to this diagram, AML is the best system in comparison with others. As expected, SANOM does have not a good performance because of lack of a translator, but its performance is slightly better than XMap and LogMapLite.

2.3. The Conference track

The conference track consists of 21 different matching tasks coming from coupling of 6 different ontologies. There are three different mapping tasks, namely mapping only classes (M1), only properties (M2),

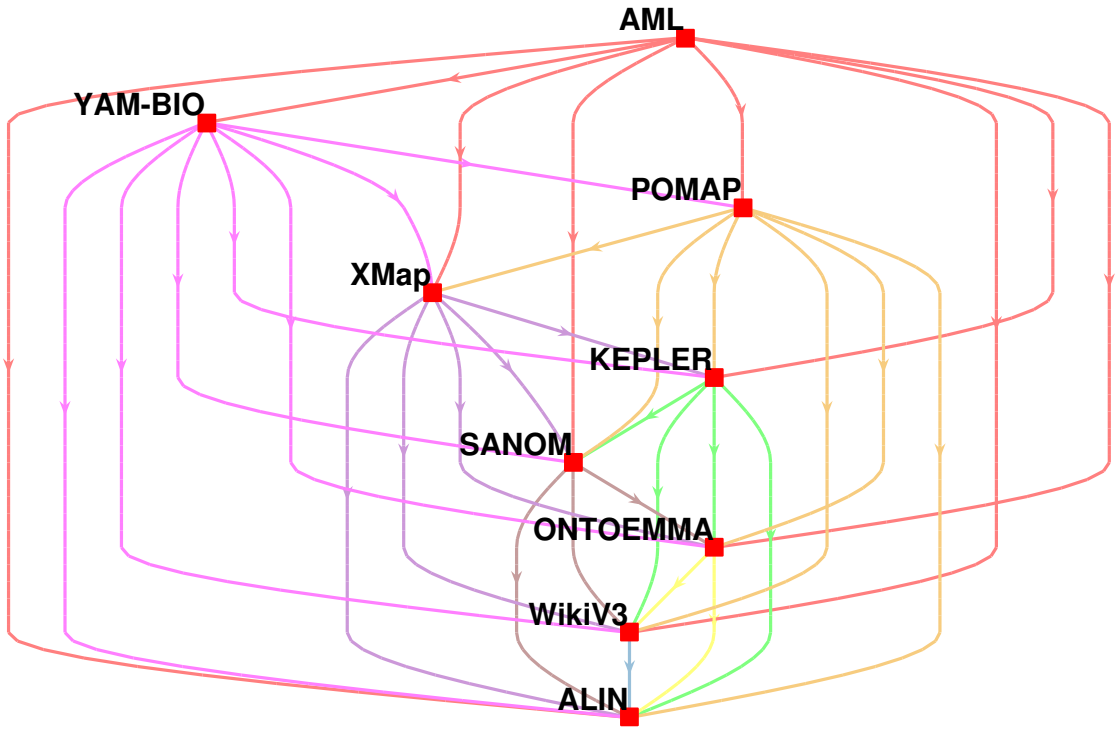


Figure 2: Comparison of alignment systems participated in OAEI 2017 on the anatomy track while the false correspondences are taken into account.

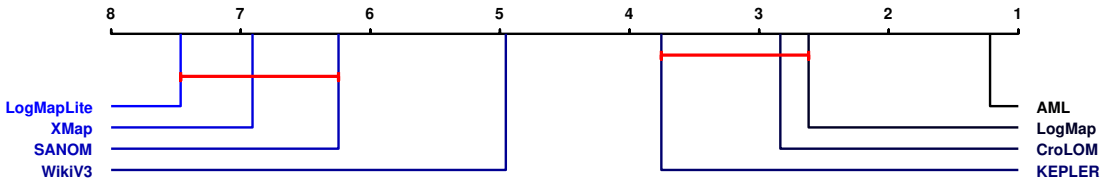


Figure 3: The critical difference diagram obtained from the Friedman test for the systems participated in the MultiFarm track. The x-axis is the rank obtained by the Friedman test and the equivalent systems are connected to each other by the red line. 3

and mapping both (M3). For the reference alignment, three different reference alignments, e.g. ra1, ra2 and rar2, are considered. Therefore, there are overally 9 different types of matching, each of which has 21 mapping tasks. Table 1 tabulates the average F-measure of SANOM in each type of matching. For the tasks which the properties is desired, SANOM has a degraded performance as its current version does not consider the matching of properties.

2.4. Conclusion

SANOM participated in OAEI 2017 for the first time. The system is in its rudimentary state, but we plan to more advance it to be able to compete with top systems. Nonetheless, the performance of SANOM is quite fair in the tracks it participated this year.

Reference

[1] M. Mohammadi, A. A. Atashin, W. Hofman, and Y. Tan, "Simulated annealing-based ontology matching," 2017.

[2] M. Mohammadi, A. A. Atashin, W. Hofman, and Y. Tan, "Comparison of ontology alignment algorithms across single matching task via the McNemar test," *arXiv preprint arXiv:1704.00045*, 2017.

[3] M. Mohammadi, W. Hofman, and Y. Tan, "A comparative study of ontology matching systems via inferential statistics," 2017.

WikiV3 results for OAEI 2017

Sven Hertling

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

Abstract. WikiV3 is the successor of WikiMatch (participated in OAEI 2012 and 2013) which explores Wikipedia as one external knowledgebase for ontology matching. The results show that the matcher is slightly better than matchers based on string equality and can get higher recall values. Moreover due to the construction of the system it is able to compute mappings in a multilingual setup.

1 Presentation of the system

1.1 State, purpose, general statement

WikiV3 is a system which exploits external knowledgebases - in this case Wikipedia. It uses the MediaWiki API and searches pages which corresponds to a given resource. When exploring the interlanguage links of Wikipedia the system is also able to find mapping between ontologies of different languages. These links point from a Wikipedia page to a correspondent page in Wikipedia with a different language. In contrast to the previous version of the matcher (WikiMatch [1] which participated in OAEI 2012 and 2013) all interlanguage links are now stored in Wikidata [1].

Wikidata is a separate project which allows to build a collaboratively edited knowledge base. One part of this project is to centralize the interlanguage links. Thus the text of Wikipedia is used to better map to Wikidata entities than just using the text available in Wikidata. The search engine of Wikipedia is based on Elasticsearch and is wrapped by a MediaWiki plugin called CirrusSearch[2]. The service provided by this plugin is heavily used by this matcher to find corresponding resources.

The general approach is shown in figure 1.

For each resource of the first ontology a list of corresponding Wikidata concepts is generated. A resource can be a class, datatype property or a object property. All of them are handled seperately to ensure that no mapping between different type of resources is generated (e.g. no class is matched to a datatype or object property). In the same way a list of Wikidata IDs (WIDs) is created for the second ontology. If there is at least one WID of a list in ontology 2 appearing in a list of WIDs in ontology 1, then a mapping is created. This will

[1] https://en.wikipedia.org/wiki/Help:Interlanguage_links
[2] https://www.mediawiki.org/wiki/Help:CirrusSearch

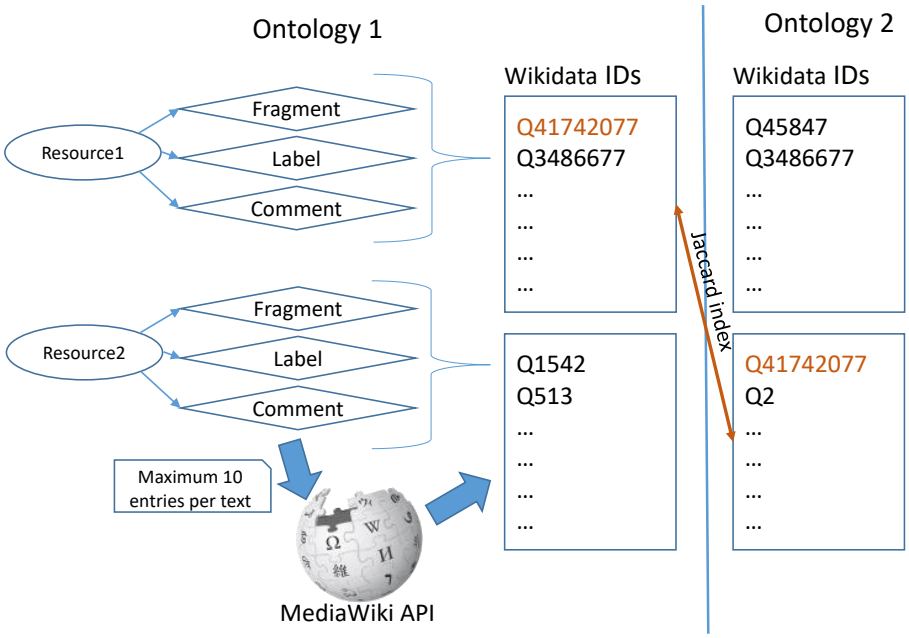


Fig. 1. Matching strategy of WikiV3

result in a n:m mapping which means one concept can be mapped to multiple other concepts. This will be reduced in a further step. The confidence value of a generated mapping is computed by the Jaccard index which is defined as

$$confidence(M) = \frac{WID(Ont1(M)) \cup WID(Ont2(M))}{WID(Ont1(M)) \cap WID(Ont2(M))} \quad (1)$$

where M represents the mapping, Ont1 and Ont2 selects the corresponding resource in Ontology one or two and the function WID returns the set of all Wikidata IDs for the corresponding resource.

The retrieval of WIDs for one resource is now described in more detail. The goal is to generate a list of WIDs which represents a given resource. In the best case there is a WID which directly represents the resource but most of the time there will be only Wikidata entries which partially represents the concept. For achieving that goal, the search API of Wikipedia is used[3].

We queried the search API for all labels, comments and for the fragment of the URI for each resource. The text length is reduced in case it is longer than 300 characters because otherwise the endpoint do not process the query. Furthermore we do not consult the endpoint if 50% of the characters are numbers. Due to the fact that the search endpoint is sensitive to tokenization (compare results from

[3] `https://www.mediawiki.org/wiki/API:Search`

"Review_preference"[4] and "Review preference"[5]), the text is tokenized (using the following characters as a splitting point:",;:()?!._- "). Afterwards all tokens are joined with a single whitespace.

The search URI[6] is parameterized and the language variable is replaced with the ISO 639-1 language code of the literal. In case there is no language tag the default language of the ontology is used (the most used language of all literals). The variable text is replaced with the processed string of the literal. With this query the suggestions of Wikipedia are also explored. Thus misspellings can be detected and fixed.

The results of this API call are Wikipedia page titles. These are converted to WIDs by using the page properties call[7] and the remaining variable joinedTitles is replaced with the Wikipedia page titles. For faster processing all queries are cached.

After comparing the WID lists from each ontology the result is a n:m mapping of the concepts with a computed confidence value which is used in a second step to increase the precision of the matcher. This step will filter all mappings below a given threshold. There are two different thresholds depending if the matching task is multilingual or not. This is detected through the default languages of both ontologies. If they differ then the threshold is not applied because in a multilingual setup the recall would drop drastically. In monolingual setup we choose a threshold of 0.28 which means that more than a quarter of the WIDs of two resources have to match.

The confidence filter does not ensure that we get a 1:1 mapping. Therefore an additional cardinality filter is applied. In case there is an n:m mapping it chooses the one with the best confidence score. As a last step all mappings which do not have the same host URI as the majority of the ontology will be deleted. This ensures that the final mapping does not contain trivial mappings.

1.2 Specific techniques used

The main technique is the usage of Wikipedia API as an external source to find mappings in Wikidata. With this information it is possible to also deal with a multilingual ontology matching setup. The filter steps of the postprocessing ensures a 1:1 mapping which is generally applicable.

1.3 Adaptations made for the evaluation

The only adaption of the system is the threshold setting. In a multilingual setup the threshold is not applied whereas in all other cases a value of 0.28 is used. In

[4] `http://en.wikipedia.org/w/index.php?search=Review_preference`

[5] `http://en.wikipedia.org/w/index.php?search=Review+preference`

[6] `https://{language}.wikipedia.org/w/api.php?action=query&list=search&format=json&srsearch={text}&srinfo=suggestion&srlimit=10&srprop=&srwhat=text`

[7] `https://{language}.wikipedia.org/w/api.php?action=query&prop=pageprops&format=json&titles={joinedTitles}&ppprop=wikibase_item`

context of the matching system this value represents the overlap in percentage of two sets consisting of WIDs representing a resource.

1.4 Link to the system and parameters file

The WikiV3 tool can be downloaded from
`https://www.dropbox.com/s/kqthgvci2onj472/WikiV3.zip`.

2 Results

2.1 Anatomy

WikiV3 has by far the highest runtime due to Wikipedia API calls (nearly 37 minutes). In comparison to the string equivalence base line the system has only a little bit higher F-measure (+0.036) but a better recall (+0.112).

The system is able to match the follwing resources but only with a low threshold.

Table 1. True positive matches in Anatomy

left label	confidence	right label
osseus spiral lamina	0.2857	Lamina_Spiralis_Ossea
thoracic vertebra 9	0.3333	T9_Vertebra
trigeminal V spinal sensory nucleus	0.3333	Nucleus_of_the_Spinal_Tract_of_the_Trigeminal_Nerve
zygomatic bone	0.3333	Zygomatic_Arch
lumbar vertebra 2	0.3333	L2_Vertebra
nasopharyngeal tonsil	0.3333	Pharyngeal_Tonsil
endocrine pancreas secretion	0.3636	Pancreatic_Endocrine_Secretion
synovium [8]	0.4000	Synovial_Membrane
xiphoid cartilage [9]	0.4286	Xiphoid_Process

If the text is more and more equal then the confidence will also arise. But these examples can be clearly also found by string comparison approaches [3].

2.2 Conference

In conference track the situation is same as in anatomy. WikiV3 is slightly better than the string equivalence baseline (+0.02 F-measure in ra1-M1). Nevertheless it finds correspondences like `http://iasted#Sponsor` = `http://sigkdd#Sponzor` (different spelling) and `http://iasted#Student_registration_fee` = `http://sigkdd#Registration_Student` (different fragment text).

[8] `https://en.wikipedia.org/wiki/Synovial_membrane`
[9] `https://en.wikipedia.org/w/index.php?search=xiphoid+cartilage&title=Special:Search`

2.3 Multifarm

In the interesting case of matching different ontologies in different languages our system achieves 0.25 F-measure. Most problematic is the recall of 0.25 because we already reduced the threshold in a multilingual setup. In most cases the concept at hand is not represented as its own Wikipedia article. Nevertheless the system is able to find mappings (exemplary for english-german) like

Table 2. True positive matches in Multifarm

left label	right label
Autor@de	author@en
Konferenz@de	conference@en
hat E-Mailadresse@de	has email@en
Dokument@de	document@en

3 General comments

3.1 Comments on the results

The overall results shows that WikiV3 is able to beat at least the string equivalence matching approaches in terms of F-measure. The recall values are higher than the one of the baselines but could be even higher.

The main drawback of the system is that most of the resources in the ontologies are not described by exactly one concept in Wikipedia (and thus Wikidata). Furthermore the Elasticsearch cluster can only deal with small misspellings and not with semantic equivalent terms or more sophisticated approaches like rewriting the query or applying any machine learning approaches. But this allows reproducible results when fixing a specific version of the cirrussearch dumps.

3.2 Discussions on the way to improve the proposed system

One improvement concern the runtime of WikiV3. Each call to Wikipedia API costs a lot of time. For a future version of this matcher it would be possible to replicate the cirrussearch dumps [10] with the given setting[11] and mapping[12] files. Querying this Elasticsearch cluster is also possible due to the ability to retrieve the corresponding query [13]. With this information a in-depth analysis

[10] `https://dumps.wikimedia.org/other/cirrussearch/`

[11] `https://en.wikipedia.org/w/api.php?action=cirrus-settings-dump&formatversion=2`

[12] `https://en.wikipedia.org/w/api.php?action=cirrus-mapping-dump&formatversion=2`

[13] `https://en.wikipedia.org/w/index.php?title=Special:Search&cirrusDumpQuery=&search=cat+dog+chicken`

of the results are feasible. This setup enables a change of the index settings and preprocessing steps to further improve the results.

In the classification of elementary matching approaches [2] the system works at the syntactic element-level and do not use any graph or model based techniques. This is a desired property for this matching system but it can be extended to also use structural information.

4 Conclusions

In this paper we analyzed the results for WikiV3 - an ontology matching system which explores Wikipedia as an external knowledge base. It is able to find more correspondences than a simple string comparison approach. Nevertheless it is only slightly better than that in terms of F-measure. Thus such a mapping approach can be used as a intermediate step to increase the recall also in multilingual setups.

References

1. Hertling, S., Paulheim, H.: Wikimatch - using wikipedia for ontology matching. In: Ontology Matching : Proceedings of the 7th International Workshop on Ontology Matching (OM-2012) collocated with the 11th International Semantic Web Conference (ISWC-2012). vol. 946, pp. 37–48. RWTH, Aachen (2012), http://ub-madoc.bib.uni-mannheim.de/33071/
2. Shvaiko, P., Euzenat, J.: A survey of schema-based matching approaches. In: Spaccapietra, S. (ed.) Journal on Data Semantics IV, Lecture Notes in Computer Science, vol. 3730, pp. 146–171. Springer Berlin Heidelberg (2005)
3. Zhou, L., Cheatham, M.: A replication study: understanding what drives the performance in wikimatch. In: Ontology Matching : Proceedings of the 12th International Workshop on Ontology Matching collocated with the 16th International Semantic Web Conference (ISWC-2017) (2017), to appear

XMap : Results for OAEI 2017

Warith Eddine DJEDDI[a,b], Mohamed Tarek KHADIR[a] and Sadok BEN YAHIA[b]

[a]LabGED, Computer Science Department, University Badji Mokhtar, Annaba, Algeria
[b]Faculty of Sciences of Tunis, University of Tunis El-Manar, LIPAH-LR 11ES14, 2092, Tunisia
{djeddi,khadir}@labged.net
sadok.benyahia@fst.rnu.tn

Abstract. We describe in this paper the XMap system and the results achieved during the 2017 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.

1 Presentation of the system

XMap, as for eXtended Mapping, is one of the leading ontology matching systems for large-scale ontology matching relying on the notion of context in order to deal with lexical ambiguity as well as a divide-and-conquer approach to tackle the issue of matching large ontologies.

In XMap, the measurement of lexical similarity in ontology matching is performed using a synset, defined in WordNet [1] and UMLS [2]. In our approach, the similarity between two entities of different ontologies is evaluated not only by investigating the semantics of the entities names, but also taking into account the context, through which the effective meaning is described. The translation into many languages is based on the Microsoft ®Translator. Our system stores locally all translation results from Microsoft ®Translator in dictionary files. The translator will also be queried only when no stored translation are found in order to gain time and avoid overloading the server.

2 State, purpose, general statement

XMap using an oracle by modifying the validation process of the candidate mappings according to the quality of the interactive matching in terms of F-measure and number of required interactions. This process is performed after each round of candidate retrieving. Our approach is based on semantic techniques and on a parallel execution strategy adapted from [3], to address the challenge of scalability and efficiency of matching techniques. One of the main trusts of the introduced approach is the increasing scalability and speed of ontology alignment by matching linguistic and structural features.

At a glance, the mapping process of XMap is depicted in Figure 1. 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. Afterwards, the alignments from all matchers can be aggregated to obtain a final alignment through the use of sequential composition [4]. Finally, a fast repair method is applied

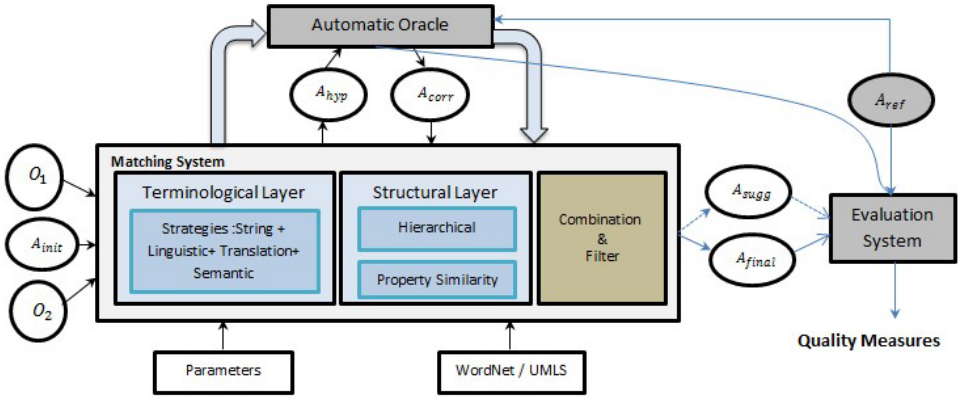


Fig. 1. The different steps for scoring a multiple network alignment.

so as to detect and remove the inconsistent classes by "'Applying Logical Constraints on Matching Ontologies"' (ALCOMO) [5]. The main goal is to try to remove less unsatisfiable classes (discovering disjointness relationships) without having an impact on the F-measure score.

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* and *Disease and Phenotype* 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.) (see Table 1). In terms of F-Measure/runtime, XMap is ranked 2nd among the tools participated in this track.

Table 1. Results for Anatomy track.

System	Precision	F-Measure	Recall	Time(s)
XMap	0.926	0.893	0.863	37

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	0.84	0.73	0.64
ra1-M2	0.75	0.32	0.2
ra1-M3	0.84	0.68	0.57
Entailed reference alignment (ra2)			
ra2-M1	0.79	0.67	0.58
ra2-M2	0.83	0.35	0.22
ra2-M3	0.79	0.63	0.52
Violation reference alignment (rar2)			
rar2-M1	0.78	0.68	0.6
rar2-M2	0.83	0.35	0.22
rar2-M3	0.78	0.65	0.55
Uncertain reference alignments (Sharp)			
-	0.84	0.57	0.6

Table 3. Results based on the uncertain version of the reference alignment.

Precision	F-Measure 1	Recall
Uncertain reference alignments (Sharp)		
0.84	0.68	0.57
Uncertain reference alignments (Discrete)		
0.79	0.72	0.67
Uncertain reference alignments (Continuous)		
0.81	0.73	0.67

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.

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.

Interactive matching evaluation The goal of this evaluation is to imitate interactive alignment [6, 7], where a oracle user is involved to validate the correspondences found

Table 4. Results for Multifarm track.

System	Different ontologies			Same ontologies		
	P	F	R	P	F	R
XMap	0.24	0.06	0.04	0.66	0.10	0.06

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 2017 edition, participating systems are evaluated on the Conference, Anatomy, Large biomedical and Phenotype 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 two years 2016 and 2017, XMap preserved roughly the same F-Measure value, and it benefits the least from the interaction with the oracle. All 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 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	0.901	0.937	20
Whole FMA-NCI	0.884	0.847	0.865	130
Small FMA-SNOMED	0.974	0.847	0.906	62
Whole FMA- Large SNOMED	0.774	0.843	0.807	625
Small SNOMED-NCI	0.894	0.566	0.693	106
Whole SNOMED-NCI	0.819	0.553	0.660	563

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).

4 General comments

4.1 Comments on the results

This is the 5th time that we participate in the OAEI campaign. The official results of OAEI 2017 show that XMap is competitive with other well-known ontology matching systems in all OAEI tracks.

4.2 Comments on the OAEI 2017 procedure

As a fifth 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

In this paper, we presented the results achieved during the 2017 edition of the OAEI campaign. The 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. Christiane D. Fellbaum. *WordNet – An Electronic Lexical Database*. MIT Press, 1998.
2. Olivier Bodenreider. The unified medical language system (UMLS): integrating biomedical terminology. *Nucleic Acids Research*, 32(Database-Issue):267–270, 2004.
3. Anika Gross, Michael Hartung, Toralf Kirsten, and Erhard Rahm. On matching large life science ontologies in parallel. In *Data Integration in the Life Sciences*, pages 35–49. Springer, 2010.
4. Jérôme Euzenat and Pavel Shvaiko. *Ontology Matching*. Springer-Verlag New York, Inc., Berlin, Heidelberg, 2007.
5. Christian Meilicke. *Alignment incoherence in ontology matching*. PhD thesis, University of Mannheim, 2011.
6. 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.
7. 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.

YAM-BIO – Results for OAEI 2017

Amina Annane,[1,2] Zohra Bellahsene, and[1] Faical Azouaou,[2]
Clement Jonquet[1,3]

[1] Laboratory of Informatics, Robotics and Microelectronics of Montpellier (LIRMM)
University of Montpellier & CNRS, France
{annane,jonquet,bella}@lirmm.fr
[2] National Higher School of Informatics (ESI), Algiers, Algeria
{f_azouaou}@esi.dz
[3] Center for BioMedical Informatics Research (BMIR), Stanford University, USA

Abstract. The YAM-BIO ontology alignment system is an extension of YAM++ but dedicated to aligning biomedical ontologies. YAM++ has successfully participated in several editions of the Ontology Alignment Evaluation Initiative (OAEI) between 2011 and 2013, but this is the first participation of YAM-BIO. The biomedical extension includes a new component that uses existing mappings between multiple biomedical ontologies as background knowledge. In this short system paper, we present YAM-BIO's workflow and the results obtained in the *Anatomy* and *Large Biomedical Ontologies* tracks of the OAEI 2017 campaign.

1 Presentation of the YAM-BIO system

1.1 State, purpose, general statement

YAM-BIO may be seen as an extension of YAM++ [5] that uses existing mappings between multiple biomedical ontologies as background knowledge to enhance the matching results. The latest version of YAM++, which we reused in YAM-BIO, obtained excellent results in multiple Ontology Alignment Evaluation Initiative (OAEI) campaigns, especially in 2013 [11]. YAM++ did not participate more since then. Four years on from the last participation, our objective this year was to establish a comparison between the potential performance of a bio-customized YAM++, and state-of-the-art systems in matching biomedical ontologies.

Over last OAEI campaigns, state-of-the-art systems such as AML [7] and LogMapBio [9] used specialized background knowledge to improve their results. More generally, the use of background knowledge –or indirect matching techniques– as recently allowed to obtain better results. YAM-BIO is an equivalent evolution of YAM++ in which we added a component that uses existing mappings as background knowledge. With YAM-BIO, we participated this year to the *Anatomy* and *Large Biomedical Ontologies (Largebio)* tracks.

1.2 YAM-BIO's general alignment worklfow

As illustrated in Fig. 1, YAM-BIO's workflow contains three main steps: First, to compute direct matching between source and target ontologies using YAM++.

Second, to compose relevant existing mappings in the background knowledge for concepts not aligned during first step. Third, to compute union of the alignments produced by the two previous steps.

Direct matching with YAM++: Annotations (labels, comments, etc.) and structures of source and target ontologies are indexed as well as the context of each entity that may be a concept or a property. Then, candidate mappings with a low annotation similarity are pre-filtered. Other advanced lexical and structural similarity measures are applied on the remaining candidate mappings, before updating their similarity scores using the structure information of source and target ontologies. Finally, a threshold is dynamically computed to select the most relevant mapping candidates. For more details on each steps of the execution of YAM++, readers may refer to [5].

Indirect matching and union: During this step YAM-BIO finds mappings for the concepts that have not been matched during direct matching with YAM++. First, background knowledge existing mappings are loaded in a list of lists noted A as follows:

1. Identifiers of all concepts in the background knowledge are added to A. The identifier of a given concept is the last part of its URI, for example the identifier of the concept that has the URI http://mouse.owl#MA_0000031 is MA_0000031.
2. Each element x of A points to a list that contains identifiers of all concepts matched to x in the background knowledge.

Then, for each source concept y that is not matched yet, YAM-BIO checks if y's identifier exists in A. If yes, YAM-BIO gets the corresponding list –pointed by y– and for each element of this list, YAM-BIO verifies if itself points to a list that contains a concept identifier from the target ontology. If so, YAM-BIO derives a new mapping and adds it to the alignment produced previously by the direct matching.

1.3 Adaptations made for the OAEI campaign

The existing mappings used as background knowledge have been extracted from Uberon [10] and the Human Disease Ontology (DOID) [13]. These ontologies contain several manually edited/curated cross references to other biomedical ontologies that we may consider as mappings.

In addition, concept identifiers of the ontologies provided for the Largebio track are not the original ones, but have been replaced by their standardized preferred labels. For this reason, we have used the NCBO BioPortal's REST API [6] to replace concept identifiers within Uberon and DOID by their standardized preferred labels.

1.4 Availability

YAM++ has now a publicly accessible online prototype version [16] and is registered on Maven repositories: http://yamplusplus.lirmm.fr. YAM-BIO has not been packaged yet to be reused by others. However, the alignment set produced

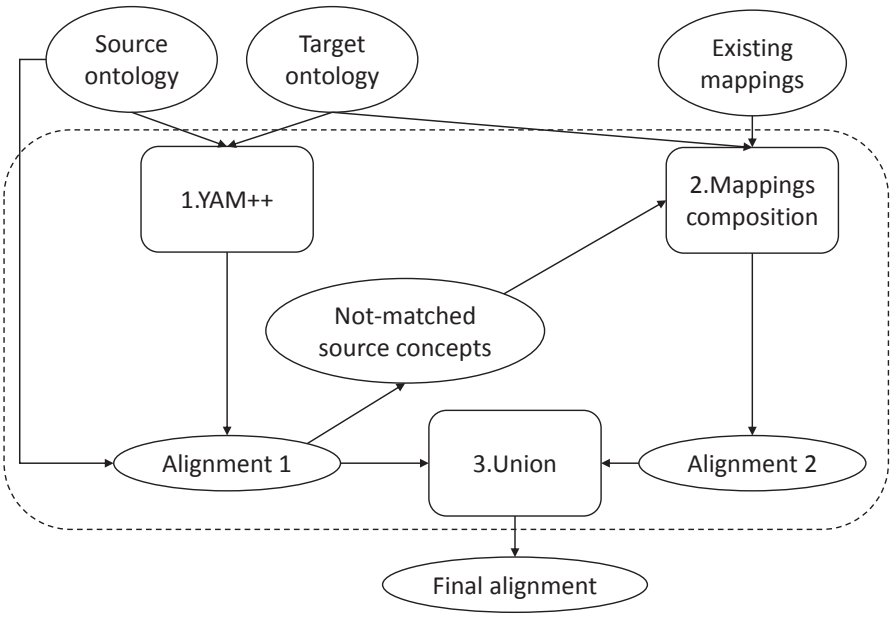


Fig. 1. YAM-BIO's general workflow

as well as the background knowledge file are available at the following link: https://goo.gl/zNznNz

2 Results

2.1 Anatomy track

The Anatomy track consists of finding an alignment between the Adult Mouse Anatomy [8] (2744 classes) and a subset of the National Cancer Institute (NCI) Thesaurus [14] (3304 classes) describing human anatomy. Table 1 shows YAM-BIO's evaluation result and runtime on this track. YAM-BIO scored in second position among the 12 systems that have participated in 2017 with almost the same precision and a slightly lower recall comparing to the top ranked system.

Table 1. YAM-BIO's Anatomy track results

Test set	Precision	Recall	F-Score	Time (s)
Anatomy	0.948	0.922	0.935	70

2.2 Large Biomedical Ontologies (Largebio) track

The Largebio track consists of respective finding alignments between the Foundational Model of Anatomy (FMA) [12], SNOMED-CT [4], and the NCI Thesaurus. There are six tasks with different input ontology sizes: small fragment, large fragment and whole ontologies. Table 2 shows YAM-BIO's evaluation re-

sults and runtime on those tasks. With the exception of the XMAP system[4], YAM-BIO is the top ranked system in Task 1 and Task 4 and obtained almost the same results as the best system in Task 3 with an F-measure of 0.834 vs 0.835. In Task 2 and Task 6, YAM-BIO scored in second position with a better recall than the best system and a lower precision. In Task 5, it shared third position with LogMapBio. In terms of running time, YAM-BIO completed the different tasks in acceptable time.

Table 2. YAM-BIO's LargeBio track results

Test set	Precision	Recall	F-Score	Time (s)
Task 1: Small fragments FMA-NCI	0.968	0.896	0.931	56
Task 2: FMA Whole-NCI Whole	0.816	0.888	0.850	279
Task 3: Small fragments FMA-SNOMED	0.966	0.733	0.834	60
Task 4: FMA Whole-SNOMED Large fragment	0.887	0.728	0.800	468
Task 5: Small fragments SNOMED-NCI	0.899	0.677	0.772	2202
Task 6: SNOMED Large fragment-NCI Whole	0.827	0.698	0.757	490

3 Discussion

3.1 Comments on the results and ways of improvement

YAM-BIO scored second position in the Anatomy track and scored first or second also in the Largebio track (except Task 5). As expected, using existing mappings as background knowledge has improved YAM++ results in terms of recall and consequently F-measure. Mapping compositions extracted from Uberon allowed YAM-BIO to discover non trivial mappings, specifically in Anatomy track and in Task 1 and Task 2 of Largebio track. Similarly, the composition of mappings extracted from DOID allowed to increase the recall of Task 5 and Task 6. However, the incoherence analysis shows that YAM-BIO returns some incoherent mappings. This may be explained by the fact that the mappings derived using background knowledge have been added to the final alignment without any semantic verification.

In our current system, mappings derived using background knowledge are not post-filtered and semantically verified as in YAM++. A simple union of the direct and indirect alignments is performed to obtain the final alignment. In the future, our goal would be to integrate the use of background knowledge directly inside YAM++'s internal architecture which, we believe, will improve coherence of the final results. More specifically, we will implement the approach proposed in [1].

In addition, we are aware of the importance of the dynamic selection of ontologies to use as background knowledge [15, 2]. Indeed, from the selected ontologies we may extract manual/automatic mappings as background knowledge. For this reason, we will extend YAM-BIO to dynamically select a set of ontolo-

[4] We note XMAP uses UMLS Metathesaurus as background knowledge, which is the same from which Largebio reference alignments are extracted.

gies from a given ontology library such as the NCBO BioPortal or Watson [3], if we want to go beyond biomedicine.

3.2 Comments on the OAEI evaluation

When possible, we think it would be interesting to publish participants results with and without use of specialized background knowledge. On one hand, this will allow to better evaluate the influence of background knowledge in matching quality and running time. On the other hand, this will allow a fair comparison with systems that do not use background knowledge.

Some components are common in all ontology matching system architectures; others do not always exist —such as background knowledge selection or semantic verification. This makes the comparison of running time executions particularly cumbersome and not always fair. According to us, it would be more appropriate to evaluate execution times for each separate component. For example, YAM-BIO used a predefined background knowledge while LogMapBio made a dynamic selection from an online repository necessarily taking additional time. Splitting running time by components will also help the community to identify less efficient components to improve them, and most efficient ones to reuse them.

4 Conclusion

In 2017 YAM-BIO participated in two tracks: Anatomy and LargeBio. The results obtained in those tracks are very close to top ranked state-of-the-art systems, thanks to different content matching techniques implemented in YAM++ and to the use of background knowledge. Due to the high heterogeneity of ontologies, we believe that an advanced generic (i.e., not restricted to biomedicine) module that selects and uses background knowledge should be implemented in the internal architecture of YAM++ to improve its results. In the future, we will work on such a module and hopefully participate in different OAEI tracks.

5 Acknowledgment

This work was done during a LIRMM-ESI collaboration within the Semantic Indexing of French biomedical Resources (grant ANR-12-JS02-01001) and PractikPharma (ANR-15-CE23-0028) projects that received funding from the French National Research Agency as well as by the European H2020 Marie Sklodowska-Curie action (agreement No 701771), the University of Montpellier and the CNRS. The authors also acknowledge the Eiffel Excellence Scholarship program.

References

1. Annane Amina, Bellahsene Zohra, Azouaou Faical, and Jonquet Clement. Selection and combination of heterogeneous mappings to enhance biomedical ontology matching. In *20th International Conference on Knowledge Engineering and Knowledge Management, EKAW, Bologna, Italy*, pages 19–33, 2016.
2. Faria Daniel, Pesquita Catia, Santos Emanuel, Cruz Isabel F, and Couto Francisco M. Automatic background knowledge selection for matching biomedical ontologies. *PloS One*, 9(11):e111226, 2014.
3. d'Aquin Mathieu, Gridinoc Laurian, Angeletou Sofia, Sabou Marta, and Motta Enrico. Watson: A Gateway for Next Generation Semantic Web Applications. In

6th International Semantic Web Conference, ISWC, Poster and Demonstration, Busan, Korea, pages 11–15, 2007.
4. Kevin Donnelly. SNOMED-CT: The advanced terminology and coding system for eHealth. *Studies in health technology and informatics*, 121:279, 2006.
5. Ngo DuyHoa and Bellahsene Zohra. Overview of YAM++:(not) yet another matcher for ontology alignment task. *Journal of Web Semantics*, 41:30 – 49, 2016.
6. Noy Natalya F, Shah Nigam H, Whetzel Patricia L, Dai Benjamin, Dorf Michael, Griffith Nicholas, Jonquet Clement, Rubin Daniel L, Storey Margaret-Anne, and Chute Christopher G. BioPortal: ontologies and integrated data resources at the click of a mouse. *Nucleic Acids Research*, 37:170–173, 2009.
7. Daniel Faria, Catia Pesquita, Booma S Balasubramani, Catarina Martins, Joao Cardoso, Hugo Curado, Francisco M Couto, and Isabel F Cruz. Oaei 2016 results of AML. In *11th International Workshop on Ontology Matching, Kobe, Japan.*, pages 138–145, 2016.
8. Terry F. Hayamizu, Mary Mangan, John P. Corradi, James A. Kadin, and Martin Ringwald. The adult mouse anatomical dictionary: a tool for annotating and integrating data. *Genome Biology*, 6(3):R29, Feb 2005.
9. E Jiménez-Ruiz, B Cuenca Grau, and V Cross. Logmap family participation in the oaei 2016. In *11th International Workshop on Ontology Matching, Kobe, Japan.*, pages 185–189, 2016.
10. Christopher J. Mungall, Carlo Torniai, Georgios V. Gkoutos, Suzanna E. Lewis, and Melissa A. Haendel. Uberon, an integrative multi-species anatomy ontology. *Genome Biology*, 13(1):R5, Jan 2012.
11. DuyHoa Ngo and Zohra Bellahsene. YAM++ results for OAEI 2013. In *8th International Workshop on Ontology Matching, Sydney, Australia.*, pages 211–218, 2013.
12. Cornelius Rosse and Jos L.V. Mejino. A reference ontology for biomedical informatics: the foundational model of anatomy. *Journal of Biomedical Informatics*, 36(6):478 – 500, 2003. Unified Medical Language System.
13. Lynn Marie Schriml, Cesar Arze, Suvarna Nadendla, Yu-Wei Wayne Chang, Mark Mazaitis, Victor Felix, Gang Feng, and Warren Alden Kibbe. Disease ontology: a backbone for disease semantic integration. *Nucleic Acids Research*, 40(D1):D940–D946, 2012.
14. Nicholas Sioutos, Sherri de Coronado, Margaret W. Haber, Frank W. Hartel, Wen-Ling Shaiu, and Lawrence W. Wright. NCI thesaurus: A semantic model integrating cancer-related clinical and molecular information. *Journal of Biomedical Informatics*, 40(1):30 – 43, 2007. Bio*Medical Informatics.
15. Chen Xi, Xia Weiguo, Jiménez-Ruiz Ernesto, and Cross Valerie. Extending an ontology alignment system with BioPortal: a preliminary analysis. In *13th International Semantic Web Conference, ISWC, Posters and Demonstrations, Riva del Garda, Italy*, pages 313–316, 2014.
16. Bellahsene Zohra, Emonet Vincent, Ngo DuyHoa, and Todorov Konstantin. Yam++ online: a multi-task platform for ontology and thesaurus matching. In *14th Extended Semantic Web Conference, ESWC, Posters and Demonstrations, Portoroz, Slovenia*, 2017.

Towards Building a Link Set Backed by Domain Experts using the Alignment Tool

Ondřej Zamazal[1], Sotirios Karampatakis[2,3], and Charalampos Bratsas[2,3]

[1] University of Economics, Prague, Dept. Information and Knowledge Engineering
ondrej.zamazal@vse.cz
[2] Aristotle University of Thessaloniki, School of Mathematics
{sokaramp@auth.gr|cbratsas@math.auth.gr}
[3] Open Knowledge Greece

1 Introduction

Discovering semantic relations between entities (entity linking) is one of the most important activity for both semantic web and linked data areas. Either we need link sets of instances or concepts we can rely on automatic systems only to a certain extent. As a result, an automatic linking is accompanied with a user interaction which enables to increase the quality of resulted link sets. Often, in order to reach as much quality of link set as possible the user should be a domain expert for an area of linking task [1]. This user specifics should be considered by designers of interactive entity linking tools. This work presents an experience from an experiment of building a link set for two fiscal code lists where domain experts have been involved. The experiment has been done using the Alignment tool.[4][5]

2 The Alignment Tool

While the Alignment tool is now a general linking tool, it has originally been developed in order to facilitate linking heterogeneous fiscal code lists in the OpenBudgets project. It is a web application for online, collaborative, system aided manual entity linking. The tool can be used to manually create link sets between two knowledge graphs or to validate already existing link sets. It further offers a number of utilities to aid the linking such as a graph visualization as a tree, a search bar, an entity description and finally suggestions based on linking algorithms provided by Silk [2] or by other automated linking tools. Multiple users can work on the same linking project simultaneously, thus enabling crowdsourcing of a link set creation and reducing required time and effort.

The user can select a semantic meaning of the link by selecting from a number of predefined link types (e.g *skos:related*,[6] *skos:broadMatch*, *owl:sameAs* etc.) or provide a custom one. The tool can also be used to crowdsource a link validation using a voting system. You can upload links produced by an automated procedure or the tool itself and create polls to check eligibility. Finally, link sets can be exported in various RDF formats and CSV.

[4] http://alignment.okfn.gr/
[5] https://github.com/okgreece/Alignment/
[6] Skos prefix refers to http://www.w3.org/2004/02/skos/core# namespace.

3 Building a Link Set by Involving Domain Experts

European union countries often apply their own different categorization systems for funded projects. As a consequence, this hinders straightforward fiscal analyses. Since there is already integrated European categorization system for funded projects, one of possible solutions to enhance fiscal analyses is to interlink categorization systems of individual EU countries to the European one. For improving this situation we started with building the one link set, the Czech code list (44 items) to the European one (142 items).[7] In order to ensure the quality of the link set we involved two domain experts and we used the Alignment tool. Thus this work enabled us testing the Alignment tool in action and examining the task of interlinking code lists with domain experts.

Our two domain experts worked separately. They followed detail guidelines[8] where they were informed about the goal of correctly interlinking as many source items to target items as possible. The guidelines also includes a brief manual how to use the Alignment tool and the instruction that experts should prefer certain types of links more, i.e. there was the following preference *skos:exactMatch*, then *skos:narrowMatch* and *skos:broadMatch* and then the others.

Both experts interlinked 32 same items where expert 1 linked 84% (37) items from the source code list and expert 2 linked 82% (36) items from the source code list. While the expert 1 employed all skos link types (out of all 53 links) more or less uniformly (21 *skos:narrowMatch*, 11 *skos:closeMatch*, 9 *skos:exactMatch*, 8 *skos:relatedMatch*, 4 *skos:broadMatch*), the expert 2 created mainly *skos:narrowMatch* links (116), additionally 8 *skos:exactMatch* and 1 *skos:broadMatch*, out of all 125 links. Both experts managed 32 times to linked the same two entities in one link and, more importantly, they managed to create the very same link 23 times where there were 7 *skos:exactMatch*, 1 *skos:broadMatch* and 15 *skos:narrowMatch*.[9] The resulted link set of 23 links represents the nucleus of the reference link set. Since there are many links created by only one expert (57% in the case of expert 1 and 82% in the case of expert 2) we further plan to let experts discuss those not agreed links to extend the current reference link set.

During the interlinking by experts we continually received a feedback in terms of bugs and improvement suggestions for the Alignment tool as also reflected via GitHub.

Acknowledgments

We thank to experts. The work has been supported by the H2020 project no. 645833.

References

1. Dragisic Z, Ivanova V, Lambrix P, Faria D, Jimnez-Ruiz E, Pesquita C. User validation in ontology alignment. In: International Semantic Web Conference 2016. Springer.
2. Volz J, Bizer C, Gaedke M, Kobilarov G. Silk-A Link Discovery Framework for the Web of Data. LDOW. 2009.

[7] Both code lists are extracted from existing data sets.
[8] The English translation is available at `https://goo.gl/vRYc5r`
[9] The further information is available at `http://owl.vse.cz:8080/OM2017/`

HOBBIT Link Discovery Benchmarks at Ontology Matching 2017

M. Röder[2,3], T. Saveta[1], I. Fundulaki[1], and A.-C. Ngonga Ngomo[2,3]

[1] Institute of Computer Science-FORTH Greece,
[2] Institute for Applied Informatics, Germany
[3] Paderborn University, Germany

Abstract. We address the problem of benchmarking ontology matching and link discovery frameworks at large scale. In particular, we aim to ensure that the benchmarks generate comparable results for the various systems and approaches. Our solution lies in implementing our benchmarks into the HOBBIT benchmarking platform, which provide means for the unified benchmarking of Big Linked Data solutions.

The HOBBIT platform serves as a framework for benchmarking Big Linked Data systems. Benchmarks that focus on the evaluation of the quality of a system using single consecutive requests can be run on the platform as well as benchmarks aiming at efficiency, e.g., by generating a lot of parallel requests leading to a high workload. Especially for the latter case, the platform supports the handling of Big Linked Data to make sure that even for high-performance systems a maximum load can be generated. The HOBBIT project[1] that designs and develops the HOBBIT platform aims at two goals: firstly, it offers an open-source evaluation platform that can be downloaded and executed locally. Secondly, it offers an online instance of the platform for a) running public challenges and b) 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 are modelled as actors with which the platform interacts. The following use cases are supported by the platform:

- *Benchmark a System:* the user selects the benchmark to test his system with. The platform loads the 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.
- *Show and Compare 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.
- *Add a System:* the user adds the system that he wants to benchmark in the platform by providing a docker image of his system and a system adapter which serves as a proxy between the benchmark and the system.

The platform can be separated into two parts. The first part comprises platform components that are always running. The second part contains all components that belong

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

to a certain experiment, i.e., the benchmark components as well as the benchmarked system. Figure 1 shows the layout of the HOBBIT platform components and how the different parts interact for the Linking and Spatial Benchmarks that are introduced in HOBBIT Link Discovery Track.[2] Below we provide a brief overview of the operation of the platform in the case of Spatial and Linking Benchmarks.[3]

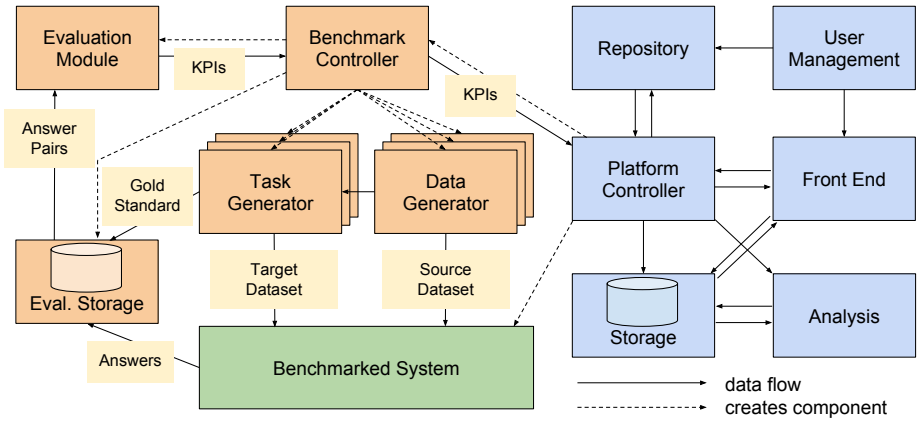


Fig. 1: Interaction of the components for the Linking and Spatial Benchmarks

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 for the experiment.

References

1. Tzanina Saveta, Irini Fundulaki, and Giorgos Flouris. Deliverable 4.1.1, first version of the linking benchmark. 2017.

[2] http://oaei.ontologymatching.org/2017/

[3] A detailed presentation of the orchestration of the different components can be found in [**?**]

Alignment: a collaborative, system aided, interactive ontology matching platform

Sotirios Karampatakis[1,2], Charalampos Bratsas[1,2], Ondřej Zamazal[3], Panagiotis Marios Filippidis[1,2], and Ioannis Antoniou[1,2]

[1] Open Knowledge Greece, Thessaloniki, Greece
http://okfn.gr
{karampatakis,cbratsas,filippidis}@okfn.gr
[2] School of Mathematics, Artistotle University of Thessaloniki, Greece
iantonio@math.auth.gr
[3] University of Economics, Prague, Dept. Information and Knowledge Engineering
ondrej.zamazal@vse.cz

1 Introduction

Ontology matching is a crucial problem in the world of Semantic Web and other distributed, open world applications. Diversity in tools, knowledge, habits, language, interests and usually level of detail may drive in heterogeneity. Thus, many automated applications have been developed, implementing a large variety of matching techniques and similarity measures, with impressive results. However, there are situations where this is not enough and there must be human decision in order to create a link[2]. In this poster we showcase Alignment platform[4][5], a novel tool developed to aid crowdsourced entity linking.

2 Alignment: The interactive, collaborative, Link Creation Web Platform

Alignment is a collaborative, system aided, user driven ontology matching platform. As previous studies have shown[1], users should not be overwhelmed with too much information, but enough in order to decide if a mapping should be created or not. With this in mind, we designed our GUI to be as minimal as can be with enough utilities to aid users, either domain or ontology engineering experts on the linking workflow. Multiple users can work on the same project and provide their own links simultaneously and interactively. The platform also offers evaluation and social features, as users can give a positive or negative vote, as well as comment on a specific link between two entities, providing feedback on the produced linksets. The produced linksets are then automatically available through both a SPARQL endpoint and an API. You can see an overview of a typical workflow in [6] A user (usually an ontology engineer) has to create a

[4] http://alignment.okfn.gr
[5] http://github.com/okgreece/Alignment
[6] https://github.com/okgreece/Alignment/blob/master/readme.md

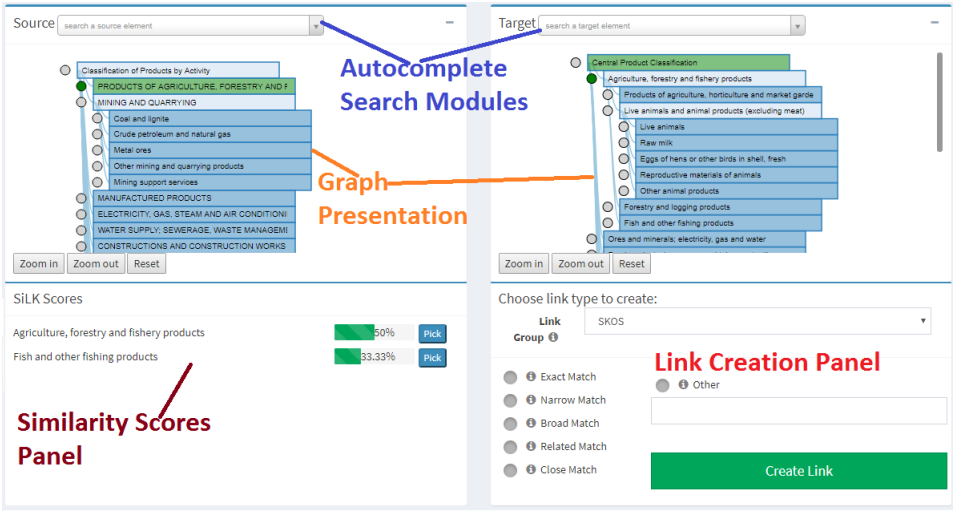


Fig. 1: Alignment GUI

project within the platform. First it is needed to upload the ontologies he wants to produce a linkset. The ontologies get validated and stored on the platform. Then the user has to define which ontology will be used as source and target ontologies consequently. Also he needs to define which similarity algorithm will be used for the system provided suggestions. The user can also choose if the project will be private or public, where multiple users can cooperate to create linksets. Then, upon creation of the project, the platform calculates similarities between the entities of the ontologies and renders the GUI. None of the suggestions provided by the system is realised as a valid link, unless some user decide to create the link. Finally, produced linksets can be exported, or send for crowdsourced validation, through the Voting service.

Acknowledgments

This work has been supported by the OpenBudgets.eu Horizon 2020 project (Grant Agreement 645833).

References

1. Dragisic, Z., Ivanova, V., Lambrix, P., Faria, D., Jiménez-Ruiz, E., Pesquita, C., Groth, P., Simperl, E., Gray, A., Sabou, M., Krötzsch, M., Lecue, F., Flöck, F., Gil, Y.: User Validation in Ontology Alignment, pp. 200–217. Springer International Publishing, Cham (2016), http://dx.doi.org/10.1007/978-3-319-46523-4_13
2. 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), http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6104044

Boosting MultiFarm Track with Turkish Dataset

Abderrahmane Khiat[1], Beyza Yaman[2], Giovanna Guerrini[2], Ernesto Jiménez-Ruiz[3], and Naouel Karam[1]

[1] Human-Centered Computing Lab, Freie Universität Berlin, Germany
[2] DIBRIS, University of Genoa, Italy
[3] University of Oslo, Norway

1 Introduction

The evolution of semantic structured data, such as those behind the deep web or social networks, requires mapping between sources to enable a high level integration. Several ontology matching systems have been developed to establish mappings between multilingual ontologies, however, employing these systems in real world requires an assessment of the ontologies capability and performance which is conducted by the MultiFarm Track. Yet, this track still lacks of ontologies from different language families. In this paper, we contribute to the OAEI initiative with a Turkish dataset to extend the coverage of languages for the matching systems.

2 The Proposed Dataset

The Turkish language comes from a different branch of the language family than the existing datasets in the Multifarm Track, we believe it will add a different perspective to the assessments of matching systems. The dataset is valuable for the OM domain for several reasons: *i)* Since the Mutltifarm track is composed of a set of ontologies of the conference domain, to the best of our knowledge no such dataset exists for the conference domain in Turkish, *ii)* we will integrate Turkish datasets to the OAEI campaign to assess the performance of cross-lingual ontology alignment systems along with other languages and *iii)* we will close the gap of lacking datasets for Turkish in the OAEI.

We followed the steps detailed in [3] to create our dataset, to validate it and then to generate the reference alignments for other ontologies. The Multifarm track has been translated from English to Turkish semi-automatically, and then reviewed and corrected by a professional English-Turkish speaker. During dataset generation, we have taken advantage of our experiences of generating Arabic datasets [2]. We first generated the Turkish ontologies via regular expressions with the regex API and, then, translated entities are replaced by the original ones. Finally, the alignments between Turkish and other languages are constructed by replacing English entity IDs by Turkish entity IDs.

However, we have a different level of difficulty for generating alignments between English-Turkish than we have experienced when we were constructing the alignments for Arabic datasets. The difficulty for English-Arabic datasets was to find an automatic solution for generating alignments between files where each one contains a high number of semantic correspondences. On the other hand, thanks to our automation of the framework, it was easier to create English-Turkish alignments by implementing a generator

for the solution. The solution consists of (1) considering the dataset that contains a high number of semantic correspondences (e.g. English-French), (2) replacing English entities by the new language (e.g. Turkish) (3) replacing entities of the other language (e.g. French) with corresponding English entities, using the alignments of the same ontologies between English and the other language (e.g. French in this case).

3 Experiments

The experimental study conducted on the Turkish datasets is performed using the CroLOM system due to its good results (ranked third) obtained in the OAEI2016 edition. CroLOM[1] uses the Yandex translator, NLP techniques and a similarity computation based on the categories of words and synonyms. The experimental results are presented in Table 1 for each language pair. The results are good for the pairs English and Spanish. However, they are less satisfactory for the pairs Chinese, Arabic and German. This is explained by the fact that CroLOM uses English as pivot to align multilingual ontologies. We can also observe that, on average Turkish ontologies bring an additional complexity to the Multifarm track, w.r.t. the results without Turkish dataset obtained via CroLOM.

Table 1: Results of the CroLOM System on the Turkish Dataset

Dataset Pairs	H-Mean Pre.	H-Mean F-mea.	H-Mean Rec.
Arabic-Turkish	0.77	0.29	0.18
English-Turkish	0.74	0.47	0.34
German-Turkish	0.59	0.36	0.26
Czech-Turkish	0.71	0.40	0.27
Chinese-Turkish	0.47	0.25	0.17
Spanish-Turkish	0.65	0.48	0.38
French-Turkish	0.60	0.42	0.33
Dutch-Turkish	0.64	0.43	0.33
Portuguese-Turkish	0.70	0.45	0.34
Russian-Turkish	0.69	0.41	0.29

The CroLOM system completed all the tests involving the Turkish language and the experimental study shows that the dataset is suitable to evaluate state-of-the-art ontology matching systems.

Acknowledgements We would like to thank Lecturer Nuriye In for her contributions to the corrections of the datasets.

References

1. A. Khiat. Crolom: cross-lingual ontology matching system results for OAEI 2016. In *Proceedings of the 11th International Workshop on Ontology Matching co-located with the 15th International Semantic Web Conference (ISWC 2016), Japan*, pages 146–152, 2016.
2. A. Khiat, G. Diallo, B. Yaman, E. Jiménez-Ruiz, and M. Benaissa. Abom and adom: Arabic datasets for the ontology alignment evaluation campaign. In *ODBASE 2015*, pages 545–553.
3. C. Meilicke, R. Garcia-Castro, F. Freitas, W. R. van Hage, E. Montiel-Ponsoda, R. R. de Azevedo, H. Stuckenschmidt, O. Sváb-Zamazal, V. Svátek, A. Tamilin, C. T. dos Santos, and S. Wang. Multifarm: A benchmark for multilingual ontology matching. *J. Web Sem.*, 15:62–68, 2012.

A Replication Study: Understanding What Drives the Performance in WikiMatch

Lu Zhou and Michelle Cheatham

DaSe Lab, Wright State University, Dayton OH 45435, USA,
{zhou.34, michelle.cheatham}@wright.edu

Abstract. We replicate and demonstrate that the performance of the WikiMatch automated ontology alignment system may be driven not by the particular information from Wikipedia directly used by the system, but rather by string similarity and Wikipedia's manually curated synonym sets, as encoded in the site's query resolution and page redirection system. In order to gain a detailed understanding of how Wikipedia contributes to WikiMatch, we replicate results reported for WikiMatch and analyze the results to evaluate our hypothesis.

1 Introduction

This paper reviews an ontology alignment system called WikiMatch. We attempt to replicate the results of the system in order to understand how Wikipedia contributes to its performance. Additionally, we conduct experiments to analyze where the performance comes from. We find that using Wikipedia can in fact find more non-syntactic pairs then using only string similarity. However, the results showed that the performance on both the conference and anatomy datasets were driven primarily by the syntactic similarity of entity labels and secondarily by the Wikipedia page redirection system.

2 Replication and Analysis

The idea behind WikiMatch is to use Wikipedia's general search functionality (through the MediaWiki API[1]) to retrieve a list of related article titles for each of the entities in the two ontologies to be aligned. After retrieving the list of titles, the similarity of each pair of entities is computed by the Jaccard index[2] on these titles. If the similarity exceeds a threshold, WikiMatch considers the entities equivalent. We began our WikiMatch replication effort by downloading the source code from the link specified in [1]. We were able to compile and run the code with minimal effort, and our results were very similar to those in the [1]. Then we used two different datasets: the conference track and anatomy track from the OAEI[3] to explore the factors driving the performance of the system.

[1] https://www.mediawiki.org/wiki/API:Search
[2] https://en.wikipedia.org/wiki/Jaccard_index
[3] http://oaei.ontologymatching.org/

Dataset	Features	Precision	Recall	F-measure	TP	FP	FN
Conference	Levenshtein String Similarity(Baseline)	0.74	0.49	0.58	150	52	155
	Directed + Redirected Queries	0.74	0.49	0.58	150	52	155
	WikiMatch(Directed + Redirected + Article Titles)	0.70	0.50	0.58	152	64	153
Anatomy	Levenshtein String Similarity(Baseline)	0.99	0.62	0.77	937	11	579
	Directed + Redirected Queries	0.99	0.62	0.77	947	11	569
	WikiMatch(Directed + Redirected + Article Titles)	0.96	0.64	0.77	966	43	550

Table 1: Comparison of different approaches on the OAEI conference (Line 1-3) and Anatomy (Line 4-6) Track (TP = True Positives, FP = False Positives, FN = False Negatives, Directed = Identical Terms with Same Title List, Redirected = Different Terms with Same Title List, Article Titles = Different Terms with Different Title List)

Table 1 shows the performance of WikiMatch compared with two other approaches to ontology alignment on two datasets. The first row of each dataset shows the performance achieved by considering two entities equivalent if their labels have a Levenstein string similarity above a threshold of 0.95. The second row shows the performance achieved by considering two entities to be equivalent if querying Wikipedia for them returns the same article. This is possible even when the entity labels are not identical because every article in Wikipedia has a *primary* term associated with it, as well as zero or more *secondary* terms that redirect to that article. For example, the primary term associated with the article on the United States of America is "United State of America", while secondary terms include "United States of America", "America", "US", and "USA". So, "United States of America" in one ontology would be found equivalent to "USA" in another ontology through this method. The final row shows the performance of the full WikiMatch system. Note that WikiMatch performs a *general search* of Wikipedia, meaning that if no article has the search term as a primary or secondary term, the search will continue over the article contents.

Overall, the percentages of correctness from string matching in the conference and anatomy dataset are 98.7% (150/152) and 98.1% (947/966) respectively. These results show that the performance of WikiMatch is mainly driven not by the article titles from Wikipedia that were used, but rather by equivalent labels string matching and the Wikipedia redirection system.

Acknowledgments This work was supported by the US Geological Survey agreement G16AC00120 "Demonstration of Semantic Web Technologies as Applied to Surface Water Feature Classification."

References

1. Hertling, S., Paulheim, H.: Wikimatch: using wikipedia for ontology matching. In: Proceedings of the 7th International Conference on Ontology Matching-Volume 946. pp. 37–48. CEUR-WS. org (2012)

Towards a complex alignment evaluation dataset

Élodie Thiéblin, Ollivier Haemmerlé, Nathalie Hernandez, Cassia Trojahn

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

Keywords: complex alignments, evaluation dataset, complex dataset

1 Motivation and background

Simple ontology alignments, largely studied, link one entity from a source ontology to one entity of a target ontology. One of the limitations of these alignments is, however, their lack of expressiveness which can be overcome by complex alignments. Different approaches for generating complex alignments have emerged in the literature [4,5,6]. However, there is a lack of datasets on which they can be evaluated.

Ontology matching is the process of generating an alignment. An alignment A between a source $o1$ and a target $o2$ ontologies is a set of correspondences [2]. Each correspondence is a triple $\langle e_{o1}, e_{o2}, r\rangle$. e_{o1} and e_{o2} are the members of the correspondence: they can be single ontology entities or constructions of these entities using constructors or transformation functions. r is a relation (e.g., $\equiv$, $\leq$, $\geq$) between e_{o1} and e_{o2}. We consider two types of correspondences:

- **simple** correspondence when both e_{o1} and e_{o2} are single entities: e.g. $\forall x$, *o1:Person*$(x) \equiv$ *o2:Human*(x) is a simple correspondence.
- **complex** correspondence when at least one of e_{o1} or e_{o2} is a construction of entities, i.e. involving at least a constructor or a transformation function. For example, $\forall$*x,y, o1:priceInDollars(x,y)* $\equiv \exists$*y1, o2:priceInEuro(x,conversion(y))* is a complex correspondence with a transformation function (*conversion* that states that $y1 = changeRate \times y$). $\forall$*x, o1:AcceptedPaper(x)* $\equiv \exists$*y, o2:Paper(x)* $\wedge$ *o2:acceptedBy(x,y)* is a complex correspondence with constructors.

A complex alignment contains at least one complex correspondence.

2 The evaluation dataset

The proposed dataset is based on the OntoFarm dataset [9] composed of 16 ontologies on the conference organisation domain and simple reference alignments between 7 of these ontologies. This dataset has been widely used in the ontology alignment evaluation domain [8]. The dataset proposed here is a first version of an extension of the OntoFarm dataset including complex correspondences. 3 out of the 7 ontologies of the reference alignments have been manually aligned (*cmt*, *conference* and *edas*), resulting in 3 alignments: *cmt-conference*, *cmt-edas* and *conference-edas*. The methodology applied to create the complex dataset consists

in manually finding an equivalent construction of target entities for each source entity. All correspondences have a single entity member and an other member that is either a single entity (simple correspondence) or a construction (complex correspondence). The correspondences are diverse for they can be classified with 8 different correspondence patterns or compositions of them [7]. In the 3 alignments, the dataset contains 51 complex correspondences. The alignments are expressed in First Order Logic and in EDOAL[1]. The resulting alignments were translated into OWL axioms as an ontology merging process. The HermiT reasoner [3] was used to check the consistency of the merged ontology. The dataset is available online at `http://doi.org/10.6084/m9.figshare.4986368.v4` under a CC-BY License.

3 Conclusion and future work

We have proposed a complex coherent dataset with complex correspondences between 3 ontologies of the OntoFarm dataset. As perspectives, the dataset will be extended with other ontologies of this dataset. The confidence of a correspondence (a value associated with a correspondence to express its confidence degree) could be added to the dataset. This could express, as in [1], the consensus level of experts on each correspondence. Finally, we aim at using this dataset for the purpose of evaluating complex matchers.

References

1. Cheatham, M., Hitzler, P.: Conference v2. 0: An uncertain version of the OAEI Conference benchmark. In: ISWC. pp. 33–48. Springer (2014)
2. Euzenat, J., Shvaiko, P.: Ontology Matching. Springer Berlin Heidelberg (2013)
3. Glimm, B., Horrocks, I., Motik, B., Stoilos, G., Wang, Z.: HermiT: An OWL 2 reasoner. Journal of Automated Reasoning 53(3), 245–269 (2014)
4. 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)
5. Parundekar, R., Knoblock, C.A., Ambite, J.L.: Linking and building ontologies of linked data. In: ISWC. pp. 598–614. Springer (2010)
6. Ritze, D., Meilicke, C., Šváb Zamazal, O., Stuckenschmidt, H.: A pattern-based ontology matching approach for detecting complex correspondences. In: 4th ISWC workshop on ontology matching. pp. 25–36 (2009)
7. Scharffe, F.: Correspondence Patterns Representation. Ph.D. thesis, Faculty of Mathematics, Computer Science and University of Innsbruck (2009)
8. Zamazal, O., Svátek, V.: 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 (Mar 2017)
9. Šváb, O., Svátek, V., Berka, P., Rak, D., Tomášek, P.: Ontofarm: Towards an experimental collection of parallel ontologies. Poster Track of ISWC 2005 (2005)

[1] `http://alignapi.gforge.inria.fr/edoal.html`

On Partitioning for Ontology Alignment

Sunny Pereira[1], Valerie Cross[1], Ernesto Jiménez-Ruiz[2]

[1] Miami University, Oxford, OH, United States, [2] University of Oslo, Norway

1 Methods

Ontology alignment (OA) for two very large ontologies becomes time consuming and memory intensive. A general approach to address these challenges is to partition each ontology into cohesive blocks. (i.e., partitions). Ontology partitioning brings new challenges: how best to partition each ontology into blocks and whether the partitioning process on each ontology should be independent of each other. In this paper, we present preliminary work to determine the suitability of partitioning strategies to improve the performance of OA systems, especially those unable to cope with the largest datasets.

The PBM (Partition Block Matching) [2,3], PAP (partition, anchor, partition) and APP (anchor, partition, partition) [1] partitioning methods have been implemented as independent methods from the alignment system. In the preliminary experiments included in this paper we report results for the systems LogMap [4] and FCA-Map [7]. In [1], [2], and [3] a path-based semantic [6] similarity measure is used to determine link strength between concepts within an ontology when creating blocks. In these experiments, the path-based Wu-Palmer [6] as well as information content based Lin [5] semantic similarity measures are considered. The ontology structure is used in determining the information content (IC) for a concept. The link strengths are calculated between concepts that only differ by one in their depth within the ontology. The authors of the PBM method use ISUB to find the anchors between concepts. In our experiments, anchors are found using an exact label match between two concepts in the two different ontologies. Each identified block pair represents a matching (sub)task, however, since blocks are only characterized by a set of concepts, they are first converted to (logical) ontology modules and then given to the ontology alignment system as input.

The initial experiments were performed on task 1 of the OAEI *largebio* track,[1] involving small fragments of FMA and NCI, using all three methods. The results using Wu-Palmer are shown below in Table 1 and those for Lin in Table 2. The parameters used are an η of 0.05 for PBM, an α of 0.75 for APP. A maximum block size of 500 and a depth difference of one for semantic similarity calculation is used for all three methods. Blocks with only one concept are considered isolated blocks. *Coverage* represents how many of the entities occurring in the OAEI reference alignments are present in the identified block pairs. The precision and recall are calculated over the combined alignment results for all the matching tasks (i.e., pair of modules extracted from the block pairs). FMA blocks (resp. NCI blocks) represents the number of total blocks produced after partitioning of the FMA ontology (resp. NCI ontology).

The results from task 1 suggest that the PBM method provides much higher recall values than the other two methods. The Wu-Palmer measure performed slightly better than Lin. The next experiments examined how the PBM with the Wu-Palmer performed on the OAEI *largebio* tasks that use the whole ontologies, that is, task 2, task 4 and task 6. The maximum block size is 3000. Table 3 presents these results.

[1] http://www.cs.ox.ac.uk/isg/projects/SEALS/oaei/

Table 1. Experiments in *largebio* task 1 suing Wu-Palmer. Matching with LogMap.

Method	FMA Blocks		NCI Blocks		Matching Tasks	Coverage	Precision	Recall	Time (s)	
	#	Isolated	#	Isolated					Partitioning	Matching
PBM	55	15	141	60	87	0.821	0.845	0.743	40.248	85.162
PAP	60	13	141	60	58	0.451	0.870	0.410	39.827	58.517
APP	50	15	143	60	48	0.518	0.870	0.472	41.644	53.157

Table 2. Experiments in *largebio* task 1 using Lin. Matching with LogMap.

Method	FMA Blocks		NCI Blocks		Matching Tasks	Coverage	Precision	Recall	Time (s)	
	#	Isolated	#	Isolated					Partitioning	Matching
PBM	46	6	180	53	83	0.801	0.833	0.728	52.454	81.689
PAP	37	5	180	53	37	0.348	0.861	0.321	56.508	39.423
APP	46	6	180	53	46	0.483	0.862	0.439	56.704	49.938

Table 3. Experiments with *largebio* whole ontologies using PBM with Wu-Palmer.

Task	System	Source Blocks		Target Blocks		Matching Tasks	Coverage	Precision	Recall	Time (s)	
		#	Isolated	#	Isolated					Partitioning	Matching
FMA-NCI	LogMap	151	2	256	91	69	0.763	0.468	0.675	649	76.7
	FCA-Map							0.506	0.698		$\approx$ 8 hrs
FMA-SNOMED	LogMap	388	9	3352	3273	154	0.594	0.571	0.423	4,807	385
SNOWMED-NCI	LogMap	3357	3160	693	427	443	0.666	0.725	0.491	6,623	937

2 Discussion and future work

In this paper we have presented a preliminary evaluation of state of the art partitioning algorithms for ontology alignment. The obtained results are not good as expected since, after the partitioning and identification of the (sub)matching tasks, the coverage of the entities in the reference alignments is rather low. For example, in the FMA-SNOMED case only 59% of the entities appearing in the reference alignment are covered by the modules in the identified matching tasks. In this case 41% of the entities were lost in either isolated blocks or blocks for which a suitable pair could not be found.

As expected, given the coverage of entities in the reference alignment, the results obtained by LogMap are very low as compared to the results reported for LogMap in last OAEI campaign. In addition the partitioning step represents a considerable overhead with respect LogMap's computation times. Nevertheless, FCA-Map was successfully run in task 2 of the *largebio* track using partitioning,[2] while the system could not cope with the task when given the whole FMA and NCI ontologies.

In the close future we aim at investigating new algorithms to provide a suitable partitioning for ontology alignment where the loss of coverage in the identified (sub)matching tasks, in terms of entities of the reference alignments, is minimized. We also intend to perform an extensive evaluation of the novel partitioning algorithms with all OAEI participating systems, especially those failing to cope with the largest tasks.

References

1. Hamdi, F., et al.: Alignment-based partitioning of large-scale ontologies. SCI, vol. 292 (2010)
2. Hu, W., Qu, Y.: Block matching for ontologies. In: Int'l Sem. Web Conf. (2006)
3. Hu, W., et al.: Matching large ontologies: A divide-and-conquer approach. DKE (2008)
4. Jiménez-Ruiz, E., Cuenca-Grau, B.: LogMap: Logic-based and scalable ontology matching. In: ISWC (2011)
5. Lin, D., et al.: An information-theoretic definition of similarity. In: ICML (1998)
6. Wu, Z., Palmer, M.: Verbs semantics and lexical selection. In: ACL (1994)
7. Zhao, M., Zhang, S.: FCA-Map results for OAEI 2016. In: Ontology Matching (2016)

[2] Not tested in tasks 4 and 6 due to limited experimental time

Paving a Research Roadmap on Network of Ontologies[1]

Fábio Santos, Kate Revoredo and Fernanda Baião

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

Abstract. Network of ontologies is the pairwise match of a set of ontologies, which became recently relevant due to its applicability in different domains, such as cultural evolution. However, the challenges faced in this area are not completely known and understood, neither are their relations to ontology matching counterpart problems. The goal of this paper is to identify challenges and applications of a network of ontologies and compare them to the 8 existing challenges of ontology matching. We identified four new challenges and related them with the eight challenges presented in [3].

Keywords: Ontology, Ontology Alignment, Network of Ontologies, Systematic Mapping Study.

1 Introducing research challenges on Network of Ontologies

After years of research and work on the Ontology field, different ontologies for describing the same domain of discourse were developed, either from scratch or based on existing ones. To deal with a number of distinct ontologies for the same domain, various Ontology matching systems were developed towards improving the process of aligning ontologies in a pair wise manner. The field of Ontology Matching evolved significantly; yet, some challenges still remain, as highlighted in [3].

With the advances on matching techniques, a network structure naturally arose, composed by the set of discovered alignments and their respective ontologies. The network environment brings new tasks that were not necessary when we were dealing with single or pairs of ontologies. So to explore the network of ontologies issues, and try to understand the relation with the matching problems, we performed a systematic mapping study and compared the challenges we have found with the challenges presented in [3]. The complete study and its results are detailed in [5].

Network of Ontologies (N.O.) is a set of ontologies with a set of alignments between them [1], or a set of theories linked by different kind of relations [2]. At some point, an application may look for alignments options in a network or a network has to be maintained with supporting tools.

We investigated, through a systematic mapping study, a research question of "*How similar is the N.O. alignment task when compared to the ontology matching*"? We have identified four challenges: network consistency detection, network revision and repair, network creation and management and inter-network matching. The first two

1 This research is partially funded by CNPq and CAPES Brazilian agencies, grant number 401505/2014-6

were mentioned in the articles selected and the last two were inferred by the N.O. definitions presented in some articles.

Figueroa *et al.* [4] presented a methodology to build ontologies and a tool to manage lifecycles. However, the approach does not address the definition of activities related to N.O. administration including user access and rights, node management, network troubleshooting and other typical activities in network environments. These problems were also not covered in [3]. If we relate the known research areas in [3] to the four "new" challenges we found after this systematic mapping, as illustrated in Figure 1, we may reveal some characteristics of the challenges.

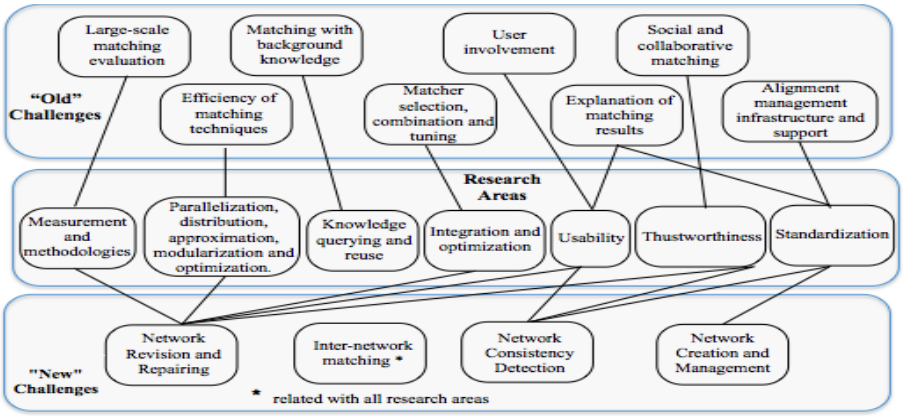


Fig. 1. Ontology Matching and N.O. challenges, and inter-related research areas

2 Final Considerations

This work provides a first roadmap for research on Network of Ontologies, by pointing a set of challenges found through the use of a systematic mapping in the literature. Interestingly, the identified challenges interrelate to previous Ontology Matching challenges posed in [3] by addressing overlapping research areas. Moreover, the new challenges found also transcend previous one with regard to specific issues such as consistency detection and alignment repair.

References

1. Euzenat, J. "Networks of ontologies and alignments." SWXO Lecture Notes (2011)
2. Euzenat, J. "Revision in networks of ontologies" Artificial intelligence 228,195-216 (2015)
3. Shvaiko, P., Euzenat, J. "Ontology matching: state of the art and future challenges." IEEE Transactions on knowledge and data engineering 25.1 pp. 158-176 (2013)
4. Suárez-Figueroa, M. C., Gómez-Pérez, A., Motta, E., & Gangemi, A., "Ontology engineering in a networked world". Springer Science & Business Media, (2012)
5. Santos, F., Revoredo, K., Baião, F., Network of Ontologies – A Systematic Mapping Study and Challenges Comparison, Technical Report. Relate-DIA/UNIRIO, RT-0005/2017, 2017. Available at http://www.seer.unirio.br/index.php/monografiasppgi/article/view/6833

Using Word Semantics on Entity Names for Correspondence Set Generation

Rafael Vieira[1] and Kate Revoredo[2]

[1,2]Federal University of the State of Rio de Janeiro (UNIRIO), Brazil,
[1]katerevoredo@uniriotec.br, [2]rvieira.research@gmail.com.br

1 Introduction

On ontology Matching, many works make use of word semantics to align the ontologies. One commonly used resource is WordNet[4][5], which groups words that share the same meaning together. Thesaurus and lexicons like WordNet indeed provide rich semantic information but require large amounts of human effort to be created and maintained.

Vector space representations of word semantics are a family of language models that associate words with vectors in a semantic space, where each dimension represents a component of the meaning of words[2][1][3]. The semantic similarity of words is exploited by these methods, providing vectors close in space when their related words are close in meaning. These vectors are usually calculated by a learning algorithm on large corpora like Wikipedia and then used to evaluate the similarity between two words.

In this work, we exploit the word-word similarities in the GloVe model as external resources for Ontology Matching. The hypothesis is that two entities can be matched based on the words in their names using the word-word similarity provided by the model. We built a prototype and evaluated its performance against the baselines from OAEI.

2 Prototype

To build the simplest prototype, we used pre-trained vectors[1] from GloVe and two ontologies O_1 and O_2. Then, each entity e defined in O_1 or O_2 is associated with one vector $\overrightarrow{v_e} = (a_1, \ldots, a_n)$, based on its name, where each component a_i represents the semantic dimension of words that have related meaning. In case entity e has a compound name, we average the vectors of each word in its name, and set the resulting vector as $\overrightarrow{v_e}$.

To generate a correspondence between two entities e_1 and e_2, from O_1 and O_2 respectively, we calculate the cosine similarity on vectors $\overrightarrow{v_1}$ and $\overrightarrow{v_2}$, associated with e_1 and e_2, respectively. If the value of cosine similarity is above a lower bound, we continue with this correspondence, otherwise, it is discarded. This lower bound was empirically set to 0.7 as this value showed the better results.

[1] Obtained at http://nlp.stanford.edu/data/glove.6B.zip

After doing this procedure for all entity pairs, we have the complete alignment. Finally, we compare this alignment with the baseline alignments edna(edit distance based) and StringEquiv(string equivalence based) from OAEI 2016 on the conference and benchmark data sets. The results are presented in table 1.

Dataset (method)	Precision	Recall	F_1-measure
Conference (edna)	0.74	0.45	0.56
Conference (StringEquiv)	0.76	0.41	0.53
Conference (Prototype)	0.71	0.45	0.54
Benchmark (edna)	0.35	0.51	0.41
Benchmark (Prototype)	0.72	0.26	0.34

Table 1. Comparison between the prototype and baselines of each data set

The prototype obtained low recall on both data sets. The majority of errors on the benchmark data set were on tests with random entity names, resulting in the low recall. This is expected since our method uses only this source of information to gather the entity semantics and then generate correspondences.

On the conference data set, the prototype performed between the two baselines. Many words from entity names were not in the vocabulary of the vectors, and were assigned the vector $\vec{0}$, which contributes to the average recall.

3 Conclusion

These results are not ground-breaking, but also promising. Furthermore, given the simplicity of the prototype, there are many places where it can be improved. For example, in a future experiment, we should train our own vectors and fine tune the hyperparameters of the model. We believe that these improvements may provide increased performance and lead to further research in the area.

References

1. Pennington, J., Socher, R. Manning, C. D.: GloVe: Global Vectors for Word Representation. Empirical Methods in Natural Language Processing (EMNLP), 1532–1543 (2014)
2. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient Estimation of Word Representations in Vector Space Computing Research Repository (CoRR), abs-1301-3781 (2013)
3. Gabrilovich, E., Markovitch, S.: Wikipedia-based Semantic Interpretation for Natural Language Processing J. Artif. Intell. Res., 34, 443–498 (2009)
4. He, W., Yang, X., Huang, D.: A Hybrid Approach for Measuring Semantic Similarity between Ontologies Based on WordNet Knowledge Science, Engineering and Management - 5th International Conference, 68–78 (2011)
5. Lin, F., Sandkuhl, K.: A Survey of Exploiting WordNet in Ontology Matching. Artificial Intelligence in Theory and Practice II, 43, 341–350 (2008)

Matching Domain and Top-level Ontologies via OntoWordNet

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

1 Context and proposed approach

Matching domain and top-level ontologies is an important task but still an open problem in the ontology matching field. The main difficulties are particularly due to their different levels of abstraction. In this paper, we propose an approach that exploits existing alignments between WordNet and top-level ontologies, as an intermediate layer, and that relies on the notion of context of concepts [1,3,5]. Contexts are constructed from all information about an ontology entity (e.g., entity naming, annotation properties and information on the neighbors of entities) and are used for disambiguating the senses that better express the meaning of ontology entities in WordNet. After selecting an appropriated synset for a given domain ontology, we verify if there is a relation between that synset and a top-level concept, via existing alignments between WordNet and the top-level ontology. Here, we focus on DOLCE top-level ontology and OntoWordNet [2]. This choice is motivated by the fact that DOLCE is one of the most used top-level ontologies and serves as a reference for the modeling and integration of ontologies [4].

2 Experiments

In order to evaluate our approach, we run experiments involving a set of 7 domain ontologies from the OAEI Conference data set[1] regarding DOLCE-Lite-Plus and OntoWordNet [2]. We focused on the first-level of domain concepts hierarchy, what corresponds to 70 concepts. This choice is motivated by the fact that correspondences can be assigned by inheritance to the child concepts. Compounds have been pre-processed and we removed the modifier (e.g. conference_document is a document). As the domain ontologies are not equipped with descriptions of their concepts, we manually enriched the first-level concepts with such definitions. For that, we adopt the Cambridge online dictionary[2] where we chosen the definition of each concept considering the most related one to the conference domain. The experiments were executed with the original and enriched versions of the domain ontologies and DOLCE-Lite-Plus (resulting in 7 pairs). This resulted in a total of 71 correspondences (including the different correspondences

[1] http://oaei.ontologymatching.org/2016/conference/index.html
[2] http://dictionary.cambridge.org/us/

found in the two versions of the domain ontologies). These 71 correspondences were presented, separately, to an expert on top-level ontologies, via an online form. The form shows the pair of concepts, their hierarchy and description. The expert was instructed to select one of the options among "equivalent", "sub concept", "none" or "other". For "other", a description of the kind of relation was required.

Results and discussion Regarding the expert judgment, 36 correspondences out of 63 for the original ontology were judged as correct. For the dictionary-enriched ontology, there are also 36 pairs considered as correct, from a total of 62. For 7 concepts in the original ontologies and 8 in the enriched ontologies, no corresponding concepts in OntoWordNet were found. Assuming that all first-level concepts in the domain ontology have potentially a corresponding concept in the top-level ontology, we compute precision, recall and F-measure. We observe similar results for both ontology versions. In fact, we expected that the descriptions would improve the synset selection and therefore produce an impact on the alignments, however the improvements were not that significant between the two versions. As we adopted plain dictionary descriptions for the terms, it might be the case that these descriptions were simply too general.

3 Concluding remarks and future work

This paper presented an approach to automatically match domain and top-level ontologies. We consider that existing top-level and WordNet alignments are 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. In addition, the precision was better than available matching systems considered in previous experiments [6]. We are aware that the experiment settings were different, but it is possibly an indication that the proposed approach might be an option for certain domains and it development should be continued and refined. As future work, we intend to improve the description of the concepts to include a more closer information about the domain, apply alternative similarity metrics for measuring the overlap between contexts, deal with logical reasoning and involve more experts in the evaluation process.

References

1. Djeddi, W.E., Khadir, M.T.: A novel approach using context-based measure for matching large scale ontologies. In: Data Warehousing and Knowl. Discovery. pp. 320–331 (2014)
2. Gangemi, A., Navigli, R., Velardi, P.: The OntoWordNet Project: Extension and Axiomatization of Conceptual Relations in WordNet, pp. 820–838. Springer Berlin Heidelberg (2003)
3. Maedche, A., Staab, S.: Measuring similarity between ontologies. In: Knowledge Engineering and Knowledge Management. pp. 251–263 (2002)
4. 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)
5. Schmidt, D., Trojahn, C., Vieira, R., Kamel, M.: Validating Top-level and Domain Ontology Alignments using WordNet. In: Brazilian Ontology Research Seminar. pp. 119–130 (2016)
6. 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)