

# Challenges of evaluating complex alignments

Beatriz Lima, Daniel Faria, and Catia Pesquita

LASIGE, Dep. Informática, Fac. Ciências da Universidade de Lisboa, Portugal

**Abstract.** The evaluation of complex ontology alignments is an open challenge, as the traditional syntactic evaluation employed for simple alignments is too unforgiving given the difficulty of accurately finding complex mappings and the usefulness of approximate solutions.

In this work we compare and discuss two simple evaluation strategies: the entity-based evaluation strategy employed in the complex track of the OAEI 2020, and a novel element-overlap-based evaluation approach we propose.

While it is clear that both strategies only provide a gross approximation of usefulness, our element-overlap strategy is the more accurate of the two, by taking semantic constructs into account. It is also more interpretable, as the final metrics are based on the total number of mappings rather than an arbitrary number of entities. Given that complex mappings often fall outside the DL spectrum, and thereby are non-decidable, a significantly more accurate measure of usefulness is not trivial.

**Keywords:** Ontology Matching · Ontology Alignment · Complex Ontology Matching · Evaluation

## 1 Introduction

Ontology alignment (or matching) emerged to overcome the semantic heterogeneity problem, by providing mappings interrelating the concepts of related ontologies [2]. While the field is well-established, most ontology alignment systems and algorithms focus exclusively on finding simple mappings connecting individual ontology entities directly through equivalence or subsumption relations [5]. However, conceptual differences between ontologies are often so profound that such simple mappings are insufficient to capture all the data transformations required for interoperability between them. Moreover, ontologies may be semantically irreconcilable through only simple mappings [4].

A complex ontology mapping is one where at least one of the mapped entities is an expression that involves multiple entities and/or logical operators or restrictions (e.g. *Accepted.contribution = min 1 acceptedBy*). Complex mappings thus enable us to express rich semantic relations between entities of two ontologies, and precisely capture the rules for converting instance data between them.

The inclusion of a complex ontology alignment track in the Ontology Alignment Evaluation Initiative<sup>1</sup> (OAEI) of 2018 [8] is an acknowledgement of the

---

<sup>1</sup> <http://oaei.ontologymatching.org>

importance of complex matching by the ontology alignment community. However, it also brought to the forefront the challenge that is providing an accurate but fair evaluation of complex ontology alignments [12].

For simple mappings, the traditional evaluation employed in most OAEI tracks—computing precision, recall and F-measure through exact match against a reference alignment—is fairly adequate. One could argue that even in this context, some mappings should be considered semi-correct, such as a subsumption mapping between two classes that are in fact equivalent, or an equivalence mapping where one of the classes is a superclass of the correct class [1]. But in practice, such cases tend to be relatively rare, and have little impact in evaluating matching systems on simple mappings.

For complex mappings the outlook is very different. First, building a complete complex reference alignment that contains all non-trivial complex correspondences is extremely laborious, which typically results in either a manual evaluation of produced mappings [6] or using partial reference alignments. Second, the intricacy of the mappings and the unbound search space (due to the nesting of expressions) mean that cases where alignment systems predict complex mappings that approximate but do not exactly match those in the reference alignment are the norm rather than the exception. Furthermore two complex mappings can be syntactically different but semantically equivalent. Thus the traditional evaluation approach is too unforgiving for complex mappings, and does not accurately reflect their usefulness [12].

A number of potential evaluation approaches have been overviewed by *Zhou et al.* [12]. They argue that an evaluation approach that reflected the expected human effort in validating a mapping, such as an edit-distance approach, would be the most suitable strategy for ontology integration tasks, predicated on the fact that manual validation would be imperative in such cases. However, to date no evaluation approach that is simultaneously automated, comprehensive and able to accurately reflect the usefulness of complex alignments has been proposed.

As of the 2020 edition, the OAEI’s complex track employs two different evaluation approaches in addition to the traditional exact match approach. In the Hydrography, Geolink and Enslaved datasets, the evaluation is an entity-based relaxed precision and recall approach which is comprehensive but neither fully automated (as the transformation of complex mappings into mapped entities is done manually) nor entirely accurate (as it doesn’t account for the semantic constructs in the complex mappings, only the entities). In the Conference and Taxon datasets, the evaluation approach is based on query answering, automatically in the case of Conference [7], but manually in the case of Taxon. This approach is able to gauge the accuracy of data transformations, but is less comprehensive than evaluation based on a full reference alignment, and requires the ability to rewrite SPARQL queries, which is still an open challenge for more expressive mappings.

In this work, we propose a novel automated element-overlap evaluation strategy for complex ontology alignments, as well as a fully automated implementa-

tion of the entity-based evaluation strategy employed in the OAEI. We assess these two strategies by using them to reevaluate the OAEI 2020 complex results, and discuss their strengths and limitations.

## 2 Related Work

The evaluation of complex ontology alignments has been comprehensively overviewed by *Zhou et al.* [12]. They detail the general framework for evaluation with a reference alignment, which consists of: **anchoring**, (mapping) **comparison**, **scoring**, and **aggregation**. Additionally, they enumerate the challenges (C) that should be addressed by an evaluation strategy:

### Anchoring

**C1:** avoid the necessity of a full pairwise comparison of reference and system mappings.

### Comparison

**C2:** determine the relation between a candidate mapping and a reference mapping.

**C3:** handle mapping decomposition (as two separate mappings can be equivalent to a single other mapping).

**C4:** factor the mapping relation.

### Scoring

**C5:** accurately reflecting the quality/usefulness of each mapping.

### Aggregation

**C6:** factor partially correct mappings.

**C7:** factor cases of mapping decomposition.

**C8:** handle the occurrence of (redundant) multiple candidate mappings that are implied by a single reference mapping.

These authors also discuss and present the challenges for evaluation without a reference alignment, namely for query answering approaches such as those employed in the OAEI. However, as these approaches are less comprehensive than evaluation with a reference alignment, we focus only on the latter.

The entity-based relaxed precision and recall approach employed in the complex track of the OAEI<sup>2</sup> begins with a manual pre-processing step, where reference and candidate mappings are converted into a list of key-value pairs of related entities plus their mapping relation. The key is a source ontology entity (or combination of entities) belonging to the mappings and manually chosen to represent them (several mappings can share the same key if they have the same source entity). The value is the set of all remaining source and target ontology entities for the mapping(s) that have the key. Considering the following mappings from *cmt – conference* task of the OAEI (including reference and hypothetical candidate mappings) as a running example:

<sup>2</sup> Unpublished; code provided by the OAEI complex track organisers

**Reference mappings:**

- (A)  $[hasDecision\ some\ Acceptance]$  or  $[\min\ 1\ acceptedBy] = Accepted\_contribution$
- (B)  $ExternalReviewer = \min\ 1\ inverseOf(invited\_by)$
- (C)  $Reviewer$  or  $ExternalReviewer = Reviewer$

**Candidate mappings:**

- (A')  $hasDecision\ some\ Acceptance > Accepted\_contribution$
- (B')  $ExternalReviewer = \min\ 1\ invited\_by$
- (C')  $Reviewer = Reviewer$

The pre-processing step would result in the following key-value pairs:

**Reference mappings:**

- (A)  $hasDecision : \{Accepted\_contribution, Acceptance, acceptedBy\}, =$
- (B)  $ExternalReviewer : \{invited\_by\}, =$
- (C)  $Reviewer : \{Reviewer, ExternalReviewer\}, =$

**Candidate mappings:**

- (A')  $hasDecision : \{Accepted\_contribution, Acceptance\}, >$
- (B')  $ExternalReviewer : \{invited\_by\}, =$
- (C')  $Reviewer : \{Reviewer\}, =$

This pre-processing step is followed by an evaluation step, where each candidate mapping is compared with the reference mapping that has the same key-entity. This comparison is done by computing the entity-precision and entity-recall of the value-entities in the candidate mapping against those in the reference mapping, and multiplying these with a relation similarity score according to the following criteria:

- 1.0 if the candidate and reference mapping have the same relation;
- 0.8 if the candidate mapping has a narrower relation (i.e.  $< vs. =, = vs. >$ );
- 0.6 if the candidate mapping has a broader relation (i.e.  $> vs. =, = vs. <$ );
- 0.3 otherwise (e.g.  $< vs. >, > vs. <$ ).

The final score of an alignment is the average of the entity scores. Applying this evaluation algorithm to the example above would result in the scores listed in Table 1.

## 3 Algorithms

### 3.1 Element-overlap-based evaluation

The element-overlap-based evaluation strategy we propose aims at gauging the expected effort to manually correct the alignment. It is based on a weighted Jaccard index between all elements of the mappings being compared (both ontology entities, and semantic constructs of the expressions) for scoring. It is not

**Table 1.** Scores obtained for the running examples under the Entity-based evaluation strategy.

Alignment	TP	FP	FN	Entity Precision	Entity Recall	Relation score	Relaxed Precision (%)	Relaxed Recall (%)
A×A'	2	0	1	1	2/3	0.6	60	40
B×B'	1	0	0	1	1	1.0	100	100
C×C'	1	0	1	1	1/2	1.0	100	50
Final	-	-	-	-	-	-	86.7	63.3

an edit-distance in the strict sense, as it captures similarity rather than dissimilarity. However, it allows us to quantify manual correction effort while reflecting mapping correctness.

Given candidate and reference complex alignments,  $A_c$  and  $A_{ref}$ , stored in data structures where each mapping is indexed by each of the ontology entities it contains, our algorithm begins with a pre-processing step, where all candidate and reference mappings are decomposed into lists of elements. For our running example, the decomposition would result in the following sets:

**Reference mappings:**

- (A)  $\{hasDecision, Acceptance, or, min, 1, accepted\_by, =, Accepted\_contribution\}$
- (B)  $\{ExternalReviewer, =, min, 1, InverseOf, invited\_by\}$
- (C)  $\{Reviewer, or, ExternalReviewer, =, Reviewer\}$

**Candidate mappings:**

- (A')  $\{hasDecision, Acceptance, >, Accepted\_contribution\}$
- (B')  $\{ExternalReviewer, =, min, 1, invited\_by\}$
- (C')  $\{Reviewer, =, Reviewer\}$

We then iterate through all candidate mappings and perform anchoring by finding related reference mappings (i.e., those that share at least one entity from both ontologies with the candidate mapping). For each related reference mapping, we compute the weighted Jaccard score between its list of elements and that of the candidate mapping. The weighted Jaccard score between two lists  $L_c$  and  $L_{ref}$  is given by:

$$WJaccard(L_c, L_{ref}) = \frac{\sum_{k \in L_c \cup L_{ref}} \min(count(k, L_c), count(k, L_{ref}))}{\sum_{k \in L_c \cup L_{ref}} \max(count(k, L_c), count(k, L_{ref}))}$$

This is an adaptation of the traditional Jaccard score between sets, taking into account that the same element can occur multiple times in a list (as is the case in a complex mapping).

We store the maximum Jaccard score found for each candidate mapping as well as for each reference mapping, which will be aggregated to compute the precision and recall respectively. Precision is computed as the average of the

best scores obtained for each mapping in the candidate alignment ( $A_c$ ), whereas Recall is the average of the best scores obtained for each mapping in the reference alignment ( $A_{ref}$ ). The scores for our running example are listed in Table 2.

The detailed description of our algorithm is provided in Algorithm 1.

**Table 2.** Element-overlap scoring for the running example mappings.

<b>Example Precision Recall</b>		
A×A'	3/8	3/8
B×B'	5/6	5/6
C×C'	3/5	3/5
Final	60.3%	60.3%

### 3.2 Automation of the OAEI entity-based pre-processing

As detailed in Section 2, the OAEI’s entity-based evaluation strategy includes a manual pre-processing step whereby reference and candidate mappings are converted into key-value pairs of related entities plus the mapping relation. The fact this step is manual obviously hinders scalability and reproducibility.

Our proposed algorithm to automate the pre-processing step of this evaluation strategy aims at emulating the manual process of identifying key-entities while operating under a set of rules that ensure an objective solution, to enable reproducibility. First, the reference alignment is converted into key-value pairs under the following rules:

1. All mappings that have a single source entity will be identified by that entity as key, and have the set of target entities as value. If more than one mapping has the same key, the values will be merged.
2. All mappings that have multiple source entities will be identified by each of the source entities that is not already the key of a single-source mapping.
  - (a) If there are multiple such source entities, the mapping will be decomposed into a key-value pair with each of those source entities as key, and the set of all target entities and all other source entities as value.
  - (b) If there are no such source entities and the mapping contains exactly two source entities, it will be identified by the set of those two source entities as key.
  - (c) If there are no such source entities and the mapping contains more than two entities, it will be identified by all pairwise combinations of source entities that are not keys of two-entity mappings.
    - i. If there are multiple such pairs of source entities, the mapping will be decomposed into a key-value pair with each of those pairs as key.
    - ii. If there is no such pair, the mapping will be identified by the set of all source entities.

Then, the candidate alignment is converted into key-value pairs using analogous rules, except that the reference alignment is used as anchor. For example, rule 2 becomes:

---

**Algorithm 1** Element-overlap evaluation algorithm

---

```

[Pre-processing]
Function convert(A)
init: HashTable lists
for mappingi in A:
    for elementj in mappingi:
        lists.add(mappingi, elementj)
End Function
init: HashTable listsref = convert(Aref),
      listsc = convert(Ac), Scoresref, Scoresc
init: double Precision = 0
for mappingi in Ac:
    [Anchoring]
    init: Ar.sources, Ar.targets
    for source_entityj ∈ mappingi:
        Ar.sources.addAll(Aref.get(source_entityj))
    for target_entityj ∈ mappingi:
        Ar.targets.addAll(Aref.get(target_entityj))
    Ar.related = Ar.sources.retainAll(Ar.targets)
    [Comparison & Scoring]
    for mappingj in Ar.related
        sim = WJaccard(listsc.get(mappingi),
                       listsref.get(mappingj))
        if sim > Scoresref.get(mappingj, sim)
            Scoresref.add(mappingj, sim)
        if sim > Scoresc.get(mappingi)
            Scoresc.add(mappingi, sim)
    [Aggregation]
    Precision += Scoresc.get(mappingi)
Precision /= Ac.size
init: double Recall = 0
for mappingi in Aref:
    Recall += Scoresref.get(mappingi)
Recall /= Aref.size

```

---

2'. All mappings that have multiple source entities will be identified by each of the source entities that is not the key of a single-source mapping **in the reference alignment**.

The same logic is applied to all rules, as the goal is to establish a parallel between the candidate alignment and the reference alignment so as to enable the evaluation of the former.

## 4 Evaluation

### 4.1 Datasets

The datasets we employed to compare the evaluation strategies were the Conference, Geolink and Hydrography datasets from the OAEI 2020 Complex track,

which are detailed in Table 3. For the Geolink and Hydrography datasets we use the reference alignment provided by the OAEI, and for the Conference dataset we employ the reference alignment provided by Thiéblin *et al.* [9], as the OAEI evaluation is query-based. We did not use the Enslaved or Taxon datasets from the OAEI 2020, because we encountered errors in the reference alignment of the former<sup>3</sup>, and no reference alignment was available for the later. We used the alignments produced by the matching systems competing in the OAEI 2020 that were able to generate complex mappings: AMLC [3], AROA [11] and CANARD [10] in the case of Geolink; AMLC and CANARD in Conference; and only AMLC in Hydrography.

**Table 3.** Description of the evaluation datasets.

Dataset	Ontologies	Alignment	Mappings	
		Tasks	Simple	Complex
Conference	5	20	111	184
GeoLink	2	1	19	48
Hydrography	4	3	113	84

## 4.2 Results and Discussion

The results of the two evaluation strategies applied to the OAEI alignments are presented in Table 4. **Element-overlap** is our proposed algorithm, **OAEI auto.** the OAEI evaluation algorithm using our automated implementation of the pre-processing step, and **OAEI man.** the OAEI evaluation algorithm with manual pre-processing, as published in the OAEI website. For the Conference dataset, the OAEI evaluation was based on query answering, which is not comparable with the two evaluation strategies, and therefore omitted from the table.

### 4.2.1 Entity-based evaluation

The results show that the OAEI entity-based evaluation with automated pre-processing closely approximates the evaluation with manual pre-processing in most cases, with the only substantial difference being observed for CANARD in the Geolink dataset. Nevertheless, it must be noted that the two variants produced exactly the same results in only one case, for AMLC on the Geolink dataset. This means that our automated implementation did not replicate all the rules that went into the manual pre-processing of the alignments, although it provided a reasonable approximation. There were likely additional criteria of a different nature (e.g. favouring classes over properties as key-entities of mappings) which we failed to identify in our analysis of the pre-processed alignments from the OAEI.

We must also note that both the OAEI entity-based pre-processing and our attempt to automate it are unnecessarily complex. Representing mappings by only key-entities, instead of simply contemplating all the entities in a mapping seems rather arbitrary, and could conceivably lead to a candidate mapping being represented by a key entity that would result in it being compared with a

<sup>3</sup> Entities in the reference alignment that were not in the ontologies.



**Table 4.** Evaluation of OAEI participating systems in the several complex datasets using our element-overlap evaluation, the OAEI entity-based evaluation using the manual pre-processing step (OAEI man.) or our automated implementation (OAEI auto.).

Alignment system	Evaluation strategy	Precision (%)	Recall (%)	F-measure (%)
<b>Conference</b>				
AMLC	Element-overlap	38±18	37±10	36±13
	OAEI auto.	49±14	38±12	42±12
CANARD	Element-overlap	24±13	43±8	29±11
	OAEI auto.	32±11	43±9	36±10
<b>Geolink</b>				
AMLC	Element-overlap	47	21	29
	OAEI man.	50	23	32
	OAEI auto.	50	23	32
AROA	Element-overlap	72	44	55
	OAEI man.	87	46	60
	OAEI auto.	88	46	60
CANARD	Element-overlap	54	33	41
	OAEI man.	89	39	54
	OAEI auto.	84	37	51
<b>Hydrography</b>				
AMLC	Element-overlap	43±15	8±10	12±14
	OAEI man.	48±17	7±8	12±13
	OAEI auto.	47±19	8±10	12±14

reference mapping that is not the most similar to it. Moreover, the result of this approach is that the precision and recall scores are neither based on the number of mappings (as in a traditional evaluation and our element-overlap approach) nor based on the total number of mapped entities (as in a pure entity-based approach), but somewhere in between, making them hard to interpret or compare. On the whole, a complete decomposition of the complex alignment into key-value pairs that encompass all mapped entities would be both more straightforward to implement and more intuitive to interpret.

#### 4.2.2 Element-overlap vs. entity-based

We can observe from the results that the entity-based evaluation is consistently more generous in terms of precision than the element-overlap-based evaluation, while recall tends to be similar for both strategies. This can be attributed to the fact that the element-overlap approach factors both the ontology entities and the semantic constructs of the expressions in its scoring, whereas the entity-based evaluation factors only the entities. Since it is generally easier to automatically find related entities than to infer the exact semantic relations between them, matching systems would tend to score higher in precision under an entity-based evaluation.

**Table 5.** Challenges addressed by our element-overlap evaluation and the OAEI entity-based evaluation.

Challenge	Element-overlap	Entity-based
(C1) Avoid full pairwise	✓	✓
(C2) Relation between mappings	✓-	✓- -
(C3 C7) Mapping decomposition	✓- -	-
(C4) Mapping relation	✓	✓
(C5) Reflect usefulness	✓-	✓- -
(C6) Partially correct mappings	✓-	✓-
(C8) Redundant mappings	-	-

An alignment accurately capturing related entities is the most critical aspect for a human reviewer, as finding which entities are related is a more time-consuming task than assessing how they are related. However, there is still a cost to the latter, which should be factored into scoring the usefulness of a mapping. As an example, consider the two reference mappings (R1, R2) from the *conference – confOf* task and the two corresponding hypothetical candidate mappings (S1, S2):

$$(R1) \textit{Reviewed.Contribution} = \min 1 \textit{InverseOf}(\textit{reviews})$$

$$(R2) \textit{Reviewer} = \min 1 \textit{reviews}$$

$$(S1) \textit{Reviewed.Contribution} = \min 1 \textit{reviews}$$

$$(S2) \textit{Reviewer} = \min 1 \textit{InverseOf}(\textit{reviews})$$

Essentially, the candidate mappings have inverted the intended usage of the *reviews* property, which would require analysis of the definition of the property to correct. Yet, under an entity-based evaluation, both candidate mappings would score 100% in precision and recall, as the presence of the *InverseOf* construct is invisible to this evaluation strategy. With our element-overlap, on the other hand, the construct would be factored into the score, providing a more accurate measure of the usefulness of the mappings.

Table 5 summarises how the two strategies address the challenges listed in Section 2. There are several challenges not addressed by our element-overlap approach, as we based it on a simple Jaccard index, knowingly sacrificing accuracy for scalability. However, there is no challenge that it addresses worse than the entity-based approach. In assessing the relation between mappings and reflecting their usefulness, it is more accurate because it takes the semantic constructs of the mappings into account. It also accounts for cases of mapping decomposition, if not very accurately, as it allows multiple candidate mappings to be compared against the same reference mapping. Since the two approaches have a similar computational cost (the pre-processing cost is much lower for the element-overlap, but the comparison cost is higher because each mapping can be compared with several other mappings), the element-overlap should be preferred.

#### 4.2.3 Jaccard vs. other similarity metrics

One limitation of our element-overlap strategy is that it doesn't factor the order in which the elements appear in a mapping, and thus would not be able to

distinguish between cases such as the following two hypothetical mappings that have the opposite meaning:

- (*Reviewer* or *ExternalReviewer*) and (not *Author*) = *Reviewer*
- not (*Reviewer* or *ExternalReviewer*) and *author* = *Reviewer*

To accurately capture such cases, we could use a canonical edit-distance approach, such as Levenshtein, but this would produce erroneous results in cases where the order of elements is irrelevant, such as within disjunction or conjunction (e.g. if *Reviewer* were swapped with *ExternalReviewer*), and would also have a higher computational cost. Another limitation of our strategy is that it cannot capture semantically related mappings (e.g. where one has a subclass or subproperty of the other) or semantically equivalent mappings that are syntactically distinct, such as these produced by CANARD for the *conference-confOf* task:

- [*contributes* and domain(*Reviewer*)] and [*reviews* and domain(*Review*)] = *reviews*
- [InverseOf(*has\_authors*) and domain(*Reviewer*)] and [InverseOf(*has\_a\_review*) and domain(*Review*)] = *reviews*

These mappings are semantically equivalent because *has\_authors* and *contributes* are inverse properties in the *conference* ontology, and so are *has\_a\_review* and *reviews*. To detect such mappings, an evaluation strategy would have to employ an OWL reasoner, which would have an even greater computational cost. Furthermore, there would be hurdles to overcome in that: (a) complex alignments are often expressed in the EDOAL format<sup>4</sup> which includes semantic constructs that aren't expressible in OWL; and (b) even OWL compliant mappings can be beyond DL semantics and therefore compromise the decidability of the reasoning problem.

Thus, while our element-overlap approach only provides a gross estimate of the usefulness of mappings, providing a significantly more accurate estimate in a scalable manner is not trivial.

## 5 Conclusion

We have proposed a novel element-overlap-based evaluation strategy for complex ontology alignments, as well as an automated pre-processing algorithm that approximates the manual pre-processing of the entity-based evaluation employed in the OAEI 2020.

We conclude that the entity-based evaluation employed in the OAEI is unnecessarily complex, and falls shorter of addressing the challenges identified for the evaluation of complex alignments [12] than our element-overlap strategy. Moreover, while our strategy knowingly sacrifices accuracy for scalability, we argue that a significant gain in accuracy is not trivial, due to complex mappings often falling outside DL semantics and thereby leading to an undecidable reasoning problem.

<sup>4</sup> <https://moex.gitlabpages.inria.fr/alignapi/edoal.html>

Nevertheless, in future work we will explore simple rule-based approaches for semantic comparison that can provide a more accurate evaluation without sacrificing scalability.

**Acknowledgements** The authors would like to thank Lu Zhou for kindly providing the source code used in the OAEI evaluation. This work was supported by FCT through the LASIGE Research Unit (UIDB/00408/2020 and UIDP/00408/2020). It was also partially supported by the KATY project which has received funding from the European Union’s Horizon 2020 research and innovation program under grant agreement No 101017453.

## References

1. Ehrig, M., Euzenat, J.: Relaxed Precision and Recall for Ontology Matching. In: Proc. K-Cap 2005 workshop on Integrating ontology. Banff, Canada, ehrig2005a
2. Euzenat, J., Shvaiko, P.: *Ontology Matching*. Springer-Verlag, Heidelberg (DE), 2nd edn. (2013)
3. Faria, D., Pesquita, C., Balasubramani, B.S., Tervo, T., Carriço, D., Garrilha, R., Couto, F.M., Cruz, I.F.: Results of AML participation in OAEI 2018. *CEUR Workshop Proceedings* **2288**, 125–131 (2018)
4. Pesquita, C., Faria, D., Santos, E., Couto, F.M.: To repair or not to repair: reconciling correctness and coherence in ontology reference alignments. *Ontology Matching* (2013)
5. Pour, N., Algergawy, A., Amini, R., Faria, D., Fundulaki, I., Harrow, I., Hertling, S., Jimenez-Ruiz, E., Jonquet, C., Karam, N., et al.: Results of the Ontology Alignment Evaluation Initiative 2020. In: *Proceedings of the 15th International Workshop on Ontology Matching (OM 2020)*. vol. 2788, pp. 92–138. CEUR-WS (2020)
6. Ritze, D., Meilicke, C., Šváb-Zamazal, O., Stuckenschmidt, H.: A pattern-based ontology matching approach for detecting complex correspondences. In: *ISWC Workshop on Ontology Matching*, Chantilly (VA US). pp. 25–36 (2009)
7. Thiéblin, E.: Do competency questions for alignment help fostering complex correspondences? In: *International Conference on Knowledge Engineering and Knowledge Management (EKAW 2018)*. pp. 1–8. Nancy, France (Nov 2018)
8. Thiéblin, É., Cheatham, M., Santos, C., Sváb-Zamazal, O., Zhou, L.: The First Version of the OAEI Complex Alignment Benchmark. In: *International Semantic Web Conference* (2018)
9. Thiéblin, É., Haemmerlé, O., Hernandez, N., Trojahn, C.: Task-oriented complex ontology alignment: Two alignment evaluation sets. In: Gangemi, A., Navigli, R., Vidal, M.E., Hitzler, P., Troncy, R., Hollink, L., Tordai, A., Alam, M. (eds.) *The Semantic Web*. pp. 655–670. Springer International Publishing, Cham (2018)
10. Thiéblin, E., Haemmerlé, O., Trojahn, C.: CANARD complex matching system: results of the 2019 OAEI evaluation campaign. vol. 2536, pp. 114–122 (2019)
11. Zhou, L., Hitzler, P.: AROA Results for 2020 OAEI. vol. 2788, pp. 161–167 (2020)
12. Zhou, L., Thiéblin, E., Cheatham, M., Faria, D., Pesquita, C., Trojahn, C., Zamazal, O.: Towards evaluating complex ontology alignments. *The Knowledge Engineering Review* **35** (2020)