On Evaluation Metrics for Complex Matching based on Reference Alignments

Guilherme Santos Sousa[1][0000−0002−2896−2362], Rinaldo Lima[2][0000−0002−1388−4824], and Cassia Trojahn[1,3][0000−0003−2840−005X]

[1] IRIT, Toulouse, France
guilherme.santos-sousa@irit.fr
[2] Universidade Federal Rural de Recife, Recife, Brazil
rjl4@cin.ufpe.br
[3] Univ. Grenoble Alpes, Inria, CNRS, Grenoble INP, France
cassia.trojahn-dos-santos@univ-grenoble-alpes.fr

Abstract. Existing metrics for evaluating complex ontology matching systems often fail to adequately capture the intricacies of (m:n) correspondences. This limitation results in partial or biased alignment quality assessments. This paper introduces a novel metric specifically tailored for complex ontology matching, extending traditional evaluation frameworks by incorporating subgraph similarity measures to ensure structural consistency with reference alignments. It utilizes a tree similarity-based approach, ensuring robustness against common issues such as order variance and detecting incorrect correspondences while adhering to key evaluation properties like completeness and correctness. Empirical experiments conducted on the OAEI complex track datasets demonstrate the superior adaptability of the metric in distinguishing correct structural correspondences compared to conventional and instance-based evaluation methods.

Keywords: Complex ontology matching · evaluation · tree similarity-based approach.

1 Introduction

Ontology matching (and more broadly, knowledge graph matching) aims to enable interoperability between knowledge expressed in different schemes. While the field has reached some maturity, most of the matching approaches still focus on generating simple (1:1) correspondences (i.e., those linking one single entity of a source ontology to one single entity of a target ontology, as $Authors \equiv Writer$). However, this type of correspondence is not expressive enough to fully cover different heterogeneities (lexical, semantic, conceptual, granularity). More expressiveness is hence fundamental and complex correspondences (i.e., those involving logical constructors or transformation functions, as e.g., $Accepted\_Paper \equiv Paper \sqcap \exists$ hasDecision.Acceptance). The need for more expressiveness has been recognized across various fields, such as cultural heritage [5], agronomic [10], or still biomedical [8, 4].

A still open issue, however, is the evaluation of complex correspondences. Existing evaluation metrics do not fully account for structural aspects, only partly exploit reference alignment, or are not always feasible. Although well-known benchmarks contain reference alignments (the result of an expert-driven curation process), no metric comprehensively compares complex correspondences from the reference with those generated by matchers. Current methods rely on manual comparison – as seen in the OAEI Taxon and Complex Conference datasets – or use partial metrics like Entity or Relationship Identification [4], as in the GeoLink benchmark. Concerning the metrics that rely on the use of common instances (A-Box data) [13], such an approach is not always feasible, as ontologies may not be exhaustively populated or may lack an A-Box altogether.

This paper introduces a novel metric specifically designed to evaluate complex ontology alignments based on a reference alignment (gold standard) while capturing their underlying semantics. Building on the evaluation framework proposed in [2], this metric addresses the shortcomings of existing approaches by fully considering the structural intricacies of (m:n) correspondences. By focusing on structural alignment, it offers a more comprehensive and accurate comparison, filling this gap in current methods. The metric assigns a higher score to the correspondence that is structurally closer to the reference, ensuring an accurate evaluation.

The structure of this paper is organized as follows. Section 2 introduces the theoretical foundations of the proposed metric, describing its desired properties, method, and algorithms. Section 3 evaluates the proposed approach through experiments, comparing its performance with state-of-the-art metrics. Section 4 provides a review of related work and existing evaluation metrics for complex ontology matching, highlighting their limitations. Finally, Section 5 summarizes the contributions of this study and outlines potential directions for future research.

2 Proposal

The work here is based on the work from [2]. Originally, precision and recall do not account for the proximity of alignments to the expected result, as they only compare exact (1:1) correspondences. This can lead to different alignments receiving the same score despite the varying quality. In that work, it is proposed to use a relaxation of the metrics by first generalizing those metrics by comparing instead the similarity between sets of correspondences and using correspondence proximity metrics, such as how distant entities are from each other, to create a fuzzy metric proximity metric. For that proposal, it is required that the same entity not appear twice in the alignment. For the evaluation of complex alignments, this restriction is incompatible, since the complex correspondences are composed of subgraphs instead of a single entity, where the same entity can appear multiple times in different subgraphs. To solve those constraints, in this

[4] https://oaei.ontologymatching.org/2020/results/complex/geolink/index.html

work is proposed to use the tree edit distance to compute the correspondence proximity and use an assignment algorithm to enable the same subgraphs to appear multiple times in the alignment. The next sections describe the theoretical foundation of the proposed metric along with its algorithmic implementation.

2.1 Foundations

Complex ontology matching involves identifying and formalizing expressive correspondences between entities in different ontologies. An alignment is defined as a set $A = \{(e_1, e_2) \mid e_1 \in O_1, e_2 \in O_2\}$, where e_1 and e_2 are entity subgraphs from the source ontology O_1 and the target ontology O_2, respectively. These pairs, (e_1, e_2), represent correspondences between semantically related elements in the two ontologies. Alignments are classified as **simple** when $|e_1| = |e_2| = 1$ (1:1) and **complex** when $\max(|e_1|, |e_2|) > 1$ (m:n).

To evaluate the quality of an alignment, a similarity function $f(A_1, A_2) \to [0, 1]$ is employed. This function measures the degree of similarity between a proposed alignment A_1 and a reference alignment A_{ref}, which acts as the gold standard. The function compares all the entity pairs $(e_1, e_2) \in A_1$ against those in A_{ref}, quantifying how well the proposed alignment captures the intended correspondences. The similarity score produced by f provides an objective metric for evaluating the alignment's adherence to the reference alignment, enabling the measurement and ranking of the matcher's performance.

2.2 Desired Properties of the Proposed Metric

Given the nature of complex alignments, a metric designed to evaluate them must comply with specific properties to ensure comprehensive assessment. For instance, a matcher that produces multiple correspondences, including both correct and incorrect ones, should be assigned a lower performance score compared to a matcher that outputs fewer but entirely correct correspondences. Based on [2], this work introduces a set of properties specifically tailored for complex matching. Unlike the original framework, which assumes that an entity can appear in only one correspondence, our work allows for the same entity to appear multiple times across different correspondences. This distinction acknowledges the fact that various logical combinations, represented as subgraphs that either include the same entity multiple times or involve distinct subgraphs containing the same entity, can convey different semantic meanings.

Considering these considerations, the desired properties of a metric for complex matching, designed to compare two sets of alignments, are as follows:

Definition 1 (Identity). $f(A_1, A_2) = 1$ *when the proposed alignment* A_1 *is identical to the reference alignment* A_2*. This ensures that the similarity function assigns the highest score when the alignments perfectly match.*

Definition 2 (Order Invariance). *Since alignments are sets of entity correspondences, the order in which they are listed should not affect the similarity*

score. Therefore, $f(A_1, A_2) = f(A_n, A_2)$*, where* A_n *is a list containing the same correspondences as* A_1 *but in a different order. This reflects that the metric compares the sets of correspondences, not their sequence. In this work, the alignments are considered to be presented as unordered lists.*

Definition 3 (Error Penalization). *The similarity score should decrease if a wrong correspondence is added to the proposed alignment. Specifically, it is expected that* $f(A_n, A_2) < f(A_1, A_2)$*, where* A_n *is* A_1 *with an additional incorrect correspondence or a redundant copy of an already correct correspondence.*

Definition 4 (Incompletness Penalization). *The similarity score should also decrease if correct correspondence is removed from the proposed alignment. Specifically,* $f(A_n, A_2) < f(A_1, A_2)$*, where* A_n *is* A_1 *without a correct correspondence. This property ensures that missing correspondences result in a lower score.*

Definition 5 (Sensitivity to Entity Modification). *Modifying an entity pair in the alignment by adding, deleting, or replacing one of its elements should also decrease the similarity score. Considering,* $f(A_n, A_2) < f(A_1, A_2)$*, where* A_n *is* A_1 *with one entity modified such that the modified pair* (e_n, e_m) *is different from the corresponding pair* (e_1, e_2) *in* A_2*. This property ensures that the metric is sensitive to changes in the specific entities or relationships in the alignment.*

2.3 Evaluation Algorithm

An evaluation algorithm is proposed to address the desired properties of the metric in the context of complex ontology alignment. This proposed algorithm receives as input an alignment file in the EDOAL format [3]. This format is commonly used to store complex alignment results between ontologies as it describes relational entities that can combine multiple entities. It is a subset of RDF/XML and its basic structure is a tree of correspondences containing cells that describe the correspondence between `edoal:entity1` tag from the source and `edoal:entity2` tag from the target with a specific confidence and relation. This algorithm does not consider the type of relation or the degree of confidence by the matcher. One example of a correspondence expressed in EDOAL is given below:

```
<Alignment>
    <map>
      <Cell rdf:about="Reviewer_merge">
        <entity1>
          <edoal:Class>
            <edoal:or rdf:parseType="Collection">
              <edoal:Class rdf:about="&cmt;Reviewer" />
              <edoal:Class rdf:about="&cmt;ExternalReviewer" />
            </edoal:or>
          </edoal:Class>
        </entity1>
```

```
        <entity2>
          <edoal:Class rdf:about="&conference;Reviewer" />
        </entity2>
        <measure rdf:datatype="&xsd;float">1.0</measure>
        <relation>Equivalence</relation>
      </Cell>
    </map>
    ...
</Alignment>
```

The choice of a tree-based algorithm is driven by the fact that EDOAL is inherently tree-structured. Although a tree is a substructure of a graph, and both graph and tree similarity algorithms could theoretically yield similar results, the tree-based approach is more natural and efficient here. While many tree and class-based similarity algorithms exist, their use in automatically evaluating complex correspondences remains unexplored.

The algorithm used to compute the TED in this work is described in [15]. In this case, the children tree is sorted as proposed in [16] since the order of the children in set operation nodes, like an intersection (owl:intersectionOf), must not impact the results. The costs used for the TED algorithm are: insertion and deletion costs are 1 and the update cost is 2. Those costs ensure that the similarity computed between trees composed of single nodes but with different entities is 0.

The proposed evaluation is computed by the Algorithm 1. The first step is loading the correspondences from the matcher output and the reference alignment in lines 2 to 3 of the Algorithm. Then the correspondences are classified into simple and complex based on the number of entities in each subtree in lines 5 to 6. Then the empty score matrix is initialized in line 7.

For each pair in the cartesian product of all correspondences between the evaluated alignment and the reference alignments, a similarity score is computed and filled into the score matrix in lines 8-14. The similarity between all correspondence pairs between A_1 and A_2 is computed as:

$$SL_{i,j} = \frac{\text{tree\_sim}(Q_{1i}, R_{1j}) + \text{tree\_sim}(Q_{2i}, R_{2j})}{2}, \quad \forall i \in \{1, \ldots, |A_1|\},\ j \in \{1, \ldots, |A_2|\} \quad (1)$$

That computation results in a matrix SL where the lines are the correspondence pairs $(Q_1, Q_2)_i$ from the source and the columns are the correspondence pairs (R_1, R_2) from the reference alignment. Now applying the assignment algorithm described in Algorithm 2.3 in line 15, a set of corresponding maps between A_1 and A_2 is retrieved that maximizes the sum of the corresponding pairs' similarities. In lines 17 and 18, the resulting assignment is classified as simple and complex into four assignments $T_{simple}, T_{complex}, Q_{simple}, Q_{complex}$. Two of them are simple assignments where the source T_{simple} and target Q_{simple} contain

Algorithm 1 Evaluate EDOAL (`evaluate_edoal`)

```
1: function EvaluateEDOAL(p1, p2, w = 0.5, sim_func = tree_sim)
2:     maps_s ← LoadAlignment(p1)
3:     maps_t ← LoadAlignment(p2)
4:     Divide alignments into simple and complex correspondences for maps_s, maps_t:
5:       S_source, C_source ← splitSimpleComplex(maps_s)
6:       S_target, C_target ← splitSimpleComplex(maps_t)
7:     scores ← Empty list for similarity scores
8:     for s1, s2 in maps_s do
9:         ms ← Empty list
10:        for t1, t2 in maps_t do
11:            ms.append((sim_func(s1, t1) + sim_func(s2, t2)) / 2)
12:        end for
13:        scores.append(ms)
14:    end for
15:    assigns ← MaximizeAssign(scores)
16:    Separate assignments into simple and complex:
17:      T_simple, T_complex ← getSourceAssignments(assigns)
18:      Q_simple, Q_complex ← getTargetAssignments(assigns)
19:    recall ← recall_avg
20:    precision ← prec_avg
21:    fmeasure ← 2 · recall · precision/(recall + precision)
22:    return soft_precision, soft_recall, soft_fmeasure
23: end function
```

simple correspondences, and the other two $T_{complex}$ and $Q_{complex}$ contain the complex assignments. Those different assignments are classified to perform different evaluations based on the number of simple or complex correspondences. After that, it is possible to compute average precision and average recall from the resulting set in lines 19 to 21.

To evaluate the performance of the matcher considering only the simple correspondences or the complex ones, a weight w is introduced. This weight ranges from 0 to 1, and when it is 0, only the simple correspondences are considered in the results, and when it's 1, only the complex correspondences are considered. The default value of 0.5 is used to evaluate both correspondence types. To perform this evaluation, assuming that $A_{source} = S_{source} \cup C_{source}$ is the set of correspondences in source and S_{source} is the set of simple correspondences in source and C_{source} is the set of complex correspondences in source. For the target, the set $A_{target} = S_{target} \cup C_{target}$ is assigned respectively. The weighted average precision is computed as:

$$prec_{avg} = \frac{(1-w) \cdot \sum_{s \in T_{simple}} s + w \cdot \sum_{c \in T_{complex}} c}{(1-w) \cdot |S_{source}| + w \cdot |C_{source}|} \tag{2}$$

And the average recall:

Algorithm 2 Maximum Assignment (`max_assign`)

```
function MaxAssign(m)
    preferences ← sorted_dict(m) ▷ Sort preferences for each pair
    unassigned ← List of unassigned pairs
    assigned ← {}
    while unassigned is not empty do
        pair ← unassigned.pop(), pair_pref ← preferences[pair]
        if len(pair_pref) = 0 then
            continue
        end if
        next_pref ← pair_pref.pop(0)          ▷ Highest remaining preference
        if next_pref[0] is in assigned then
            if next_pref[1] > assigned[next_pref[0]][1] then      ▷ Better match
                unassigned.append(assigned[next_pref[0]][0])      ▷ Free current
                assigned[next_pref[0]] ← (pair, next_pref[1])
            else
                unassigned.append(pair)                           ▷ Remain free
            end if
        else
            assigned[next_pref[0]] ← (pair, next_pref[1])         ▷ Assign the pair
        end if
    end while
    return assigned
end function
```

$$recall_{avg} = \frac{(1-w)\cdot\sum_{s\in Q_{simple}} s + w\cdot\sum_{c\in Q_{complex}} c}{(1-w)\cdot|S_{target}| + w\cdot|C_{target}|} \tag{3}$$

Also, an aggregated metric such as f-measure can be computed with the averaged precision and recall:

$$f1 = \frac{2\cdot prec_{avg}\cdot recall_{avg}}{prec_{avg} + recall_{avg}}, (prec_{avg} + recall_{avg}) \neq 0 \tag{4}$$

The proposed evaluation method in this paper is implemented in Python and is available at GitLab[5].

2.4 Properties Verification

In this section, the arguments for the properties of the proposed evaluation method are stated.

Lemma 1 (Identity). *Given alignments A_1 and A_2, if $A_1 = A_2$, then the similarity function $f(A_1, A_2) = 1$.*

[5] https://gitlab.irit.fr/melodi/ontology-matching/complex/complex-reference-evaluation

Proof. Assume that $A_1 = A_2$. From the definition of the similarity metric, $f(A_1, A_2)$ is computed using the matrix sl of pairwise similarities. Since $A_1 = A_2$, all entity pairs (m_1, m_2) compared in sl will yield $tree\_sim(m_1, m_2) = 1$ when $m_1 = m_2$ and lower similarity for all other pairs. The assignment algorithm $maximize\_assign$ will therefore select matches that maximize the total similarity, which will select the equal pairs. Since all selected pairs have similarity 1, precision and recall are 1, resulting in $f(A_1, A_2) = 1$.

Lemma 2 (Order Invariance). *The similarity function $f(A_1, A_2)$ is invariant to the order of entity pairs in A_1. Specifically, if A_n is a reordering of A_1, then $f(A_1, A_2) = f(A_n, A_2)$.*

Proof. Assume A_n is a reordering of A_1. Since all candidate pairs are iterated from the similarity matrix sl, even if a candidate assignment starts at a lower similarity, it will always be replaced by a higher similarity pair, leading to the same sum of similarities. If two distinct pairs, containing different entities but having the same similarity, are assigned in a different order, the total sum remains unchanged. Consequently, the precision and recall values stay consistent, regardless of the initial order.

Lemma 3 (Error Penalization). *If an incorrect correspondence $(e_1, e_2) \notin A_2$ is added to A_1 to form A_n, then $f(A_n, A_2) < f(A_1, A_2)$.*

Proof. Assume A_n is a copy of A_1 with an additional incorrect correspondence. The similarity matrix sl will stay the same as $f(A_1, A_1)$. However, the precision value in this case will be lower, leading to a lower f-measure and then $f(A_n, A_1) < f(A_1, A_1)$.

Lemma 4 (Incompletness Penalization). *If a correct correspondence is removed from A_1 to form A_n, then $f(A_n, A_2) < f(A_1, A_2)$.*

Proof. Assume A_n is A_1 with one correct correspondence removed. This reduces the number of high-similarity values in the matrix sl, resulting in a lower optimal assignment score from $maximize\_assign$. Consequently, recall and f-measure decrease.

Lemma 5 (Sensitivity to Entity Modification). *If an entity pair in A_1 is modified to form A_n, then $f(A_n, A_2) < f(A_1, A_2)$.*

Proof. Assume A_n is A_1 with one entity pair (e_1, e_2) replaced by (e_n, e_2) such that $sim(e_n, e_2) < sim(e_1, e_2)$. The modified pair will have a lower similarity score, leading to reduced values in sl. This decreases the total sum of similarity scores from $maximize\_assign$, lowering precision, recall, and f-measure.

2.5 Specific Case Exploration

To illustrate the main differences between the proposed metrics and the two common approaches, an example is drawn. As stated above, instance-based and

Entity Identification are currently used in the OAEI complex track. However, as instance-based metrics act as a proxy metric by measuring the amount of common instances returned by the correspondences, that metric is not directly comparable to the others. So, to illustrate the difference between Entity Identification and the proposed metric, consider the following example:

```
Matcher 1:
IntersectionOf( InverseOf(isWrittenBy), isAuthorOf ) = writePaper
Matcher 2:
IntersectionOf(isWrittenBy, InverseOf(isAuthorOf) ) = writePaper
Matcher 3:
UnionOf(isWrittenBy, InverseOf(isAuthorOf)) = writePaper
Reference alignment:
IntersectionOf( InverseOf(isWrittenBy), isAuthorOf ) = writePaper
```

In the OAEI results page of the year 2020 [6] (the one with the most different matchers participating), it remains unclear whether the Entity Identification task accounts for OWL predicates. In the example, if no OWL predicates are considered, since isWrittenBy and isAuthorOf appear both in the reference on the source side and writePaper on the target, all matchers score 1.0 (the maximum score considering the formula found entities / total entities). Since in the example, Matcher 1 is the exact copy of the reference alignment and the others have modifications, the ranking of the matchers doesn't reflect the desired evaluation. If the predicates are considered, the scores vary: In Matcher 1, all entities are present, so the score is 1.0. In Matcher 2, all entities are present but in a different order, yet the score remains 1.0 because Entity Identification doesn't consider the order. In Matcher 3, the inverse appears in the wrong property and, instead of Intersection, a Union is used, so this matcher scores 0.75.

In contrast, our proposed metric uses tree edit distances to measure the similarity between the generated correspondences and the reference alignment. In Matcher 1, all entities are present and in the same order, so the score is 1.0. In Matcher 2, the cost is 2 over 4 entities, so the score is 0.5. And in Matcher 3, the cost is 3 over 4 entities, reaching a score of 0.25. Thus, the proposed metric more effectively ranks correspondences based on their compliance with the reference alignment.

Another example can be considering the comparison for the entity pair illustrated in Figure 1. Using the proposed metric, four edits are required to transform the tree structure **A** into that of the tree **B**. With the total size of the compared trees being 12, the similarity score for this entity pair is computed as $1 - \frac{4}{12} = 0.33$.

By contrast, the Entity Identification process used, for example, in the populated conference evaluation in the complex track in OAEI 2020, only considers ontology-related entities, filtering the structural entities. In this case, the similarity of those trees will be 0. This evaluation presents an optimistic view of the task, simplifying the task for matchers by ignoring structural requirements; as

[6] https://oaei.ontologymatching.org/2020/results/complex/geolink/index.html

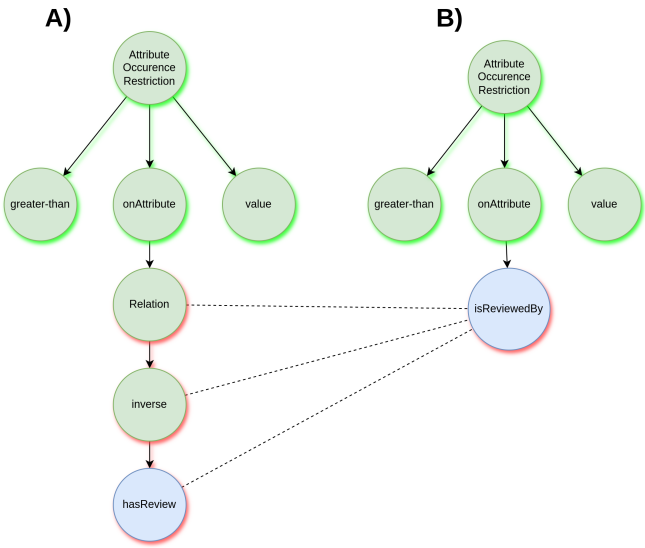

Fig. 1. Example of the similarity comparison applied in the proposed metric. Entities shaded in green are structural nodes while those in blue are ontology-specific nodes. Entities highlighted in green are correspondents, while entities highlighted in red are replaced.

long as the correct entities are included in the set, the matcher can achieve a higher score without adhering to the proper logical operators that define part of the tree structure.

Based on these observations, it is possible to outline the properties of each metric concerning the proposed definitions for a robust metric for complex matching using reference alignments. These properties are summarized in Table 1. It is possible to see that the proposed metric follows all the proposed definitions, while the relaxed-based metrics fail in property 5 and the Instance-based metrics fail in property 3 and don't apply to property 5.

Metric	Identity	Order Invariance	Error Penalization	Incompleteness Penalization	Sensitivity to Entity Modification
Proposed	✓	✓	✓	✓	✓
Entity Identification	✓	✓	✓	✓	X
Instance-based	✓	✓	X	✓	-

Table 1. Evaluation of metrics based on adherence to properties defined for proper evaluation of complex alignments. In the table, X marks when the metric doesn't follow the definition and - marks when the definition is not applied.

3 Experiments and Discussion

In this section, the results of the most commonly used automatic evaluation metrics are empirically compared. For this experiment, we analyzed the outputs of matchers that participated in the 2020[7] OAEI complex track [8]. In that evaluation, only Taxon was not included since no reference alignment is present for this track. The matchers included in this analysis are AMLC, CANARD, and AROA since those are the unique producing complex alignments in that year.

The evaluation metric proposed in this paper is compared with existing metrics. These include the instance-based evaluator introduced in [9], which is used in the Populated Conference evaluation, and Entity Identification approach used for the Geolink dataset and other datasets within the complex track [11], as well as in related works [1].

To explore the impact of different types of correspondences, the proposed metric was tested with weight values of $w = 0$, $w = 0.5$, and $w = 1.0$, representing simple correspondences, complex correspondences, and a combination of both, respectively. The results for the proposed metric are present in Table 2.

Not all matchers produced alignments for all datasets, and some only generated alignments in the simplified format designed for simple correspondences. Since the proposed metrics are constructed specifically for dealing with EDOAL, certain alignments were unsupported during the loading phase, or the parsed structure resulted in zero scores across all metrics.

When comparing simple and complex evaluation scenarios, it is possible to see that the matchers perform better in aligning simple entities but have lower performance with complex correspondences. In contrast to the proposed metric in this paper, the Entity Identification and recall evaluate the matchers in some datasets with relatively higher performance, as expected. This is because those metrics do not consider structural constraints. For example, in Populated Geolink, CANARD reaches 0.26 f-measure in the balanced proposed metric but gets 0.54 in the relaxed f-measure. But considering only complex alignments is possible to see that in this case, CANARD gets 0.09 f-measure. It gets a higher value in the relaxed metric since that metric ignores structural considerations, leading to higher result values. The same case occurs with AROA, which gets 0.39 in the Proposed metric with $w = 0.5$ and 0.24 considering only complex correspondences, but gets 0.60 in the relaxed f-measure. This highlights the importance of evaluating a system's ability to handle structural complexities (a key capability of modern AI models like LLMs) that comparison similarities fail to capture.

In the instance-based evaluation, which results are in the Populated Conference dataset in Table 2. There were no results for the Geolink, Hydrography, and Populated Enslaved dataset, as it lacks the Competency Question for Alignment (CQA) required for this evaluation. In the populated conference dataset, CANARD outperformed AMLC relative to the instance-based evaluation, but in

[7] This year was used in this experiment since in 2024 only one matcher was submitted.
[8] https://oaei.ontologymatching.org/2020/results/complex/index.html

Dataset	Metric	AMLC	CANARD	AROA
Conference	Proposed w=0.5 (Precision)	0.45	-	-
	Proposed w=0.5 (Recall)	0.19	-	-
	Proposed w=0.5 (F1)	0.26	-	-
	Proposed w=1 (Precision)	0.45	-	-
	Proposed w=1 (Recall)	0.65	-	-
	Proposed w=1 (F1)	0.51	-	-
	Precision (Manual)	0.31	-	-
	F-measure (Manual)	0.34	-	-
	Recall (Manual)	0.37	-	-
Populated Conference	Proposed w=0.5 (Precision)	0.38	0.25	-
	Proposed w=0.5 (Recall)	0.47	0.52	-
	Proposed w=0.5 (F1)	0.40	0.33	-
	Proposed w=0 (Precision)	0.69	0.60	-
	Proposed w=0 (Recall)	0.46	0.55	-
	Proposed w=0 (F1)	0.54	0.56	-
	Proposed w=1 (Precision)	0.21	0.12	-
	Proposed w=1 (Recall)	0.54	0.43	-
	Proposed w=1 (F1)	0.27	0.17	-
	Precision (Instance-based)	0.23-0.51	0.25-0.88	-
	Coverage (Instance-based)	0.26-0.31	0.40-0.50	-
Hydrography	Proposed w=0.5 (Precision)	0.02	-	-
	Proposed w=0.5 (Recall)	0.01	-	-
	Proposed w=0.5 (F1)	0.01	-	-
	Proposed w=1 (Precision)	0.05	-	-
	Proposed w=1 (Recall)	0.06	-	-
	Proposed w=1 (F1)	0.06	-	-
	Relaxed Precision	0.45	-	-
	Relaxed F-measure	0.10	-	-
	Relaxed Recall	0.05	-	-
Geolink	Relaxed Precision	0.50	-	-
	Relaxed F-measure	0.32	-	-
	Relaxed Recall	0.23	-	-
Populated Geolink	Proposed w=0.5 (Precision)	-	0.45	0.71
	Proposed w=0.5 (Recall)	-	0.18	0.27
	Proposed w=0.5 (F1)	-	0.26	0.39
	Proposed w=0 (Precision)	-	1.00	0.94
	Proposed w=0 (Recall)	-	0.78	0.82
	Proposed w=0 (F1)	-	0.88	0.87
	Proposed w=1 (Precision)	-	0.20	0.57
	Proposed w=1 (Recall)	-	0.06	0.16
	Proposed w=1 (F1)	-	0.09	0.24
	Relaxed Precision	0.50	0.89	0.87
	Relaxed F-measure	0.32	0.54	0.60
	Relaxed Recall	0.23	0.39	0.46
Populated Enslaved	Proposed w=0.5 (Precision)	0.24	0.16	-
	Proposed w=0.5 (Recall)	0.10	0.10	-
	Proposed w=0.5 (F1)	0.14	0.12	-
	Proposed w=0 (Precision)	0.24	0.18	-
	Proposed w=0 (Recall)	0.47	0.50	-
	Proposed w=0 (F1)	0.32	0.26	-
	Proposed w=1 (Precision)	0.23	0.16	-
	Proposed w=1 (Recall)	0.03	0.03	-
	Proposed w=1 (F1)	0.06	0.05	-
	Relaxed Precision	0.73	0.42	0.80
	Relaxed F-measure	0.40	0.19	0.51
	Relaxed Recall	0.28	0.13	0.38

Table 2. Comparison of metrics across different datasets and matchers. Results of zero in all metrics in some datasets were omitted for brevity. Entity Identification is a subprocess in Relaxed Precision and Recall for the results in this table.

the proposed metric, AMLC has higher results than CANARD, showing that depending on the metric, the ranking of matchers' performance changes. However, this evaluation does not distinguish between the number of simple and complex correspondences returned, making it unclear whether higher results stem from better handling of simple or complex cases. Incorporating this distinction would enhance performance analysis and provide more granular insights into the matchers' capabilities, as was done in the proposed metric. Another characteristic of the instance-based evaluator is that duplicating correspondences and adding them to the alignment does not affect the results. This can lead to issues where proposing multiple small modifications of the same correspondence will not result in reduced precision.

Another observation is the higher resource consumption of instance-based evaluations. These evaluations require CQAs as input for the source and target ontologies, along with CQAs for the related dataset, also with the system outputs. However, precision evaluation does not require CQAs. In contrast, the proposed metric only relies on the output alignment and reference alignment without the need to load the datasets. While obtaining reference alignments can be challenging, the proposed metric is compatible with six out of seven datasets in the 2020 OAEI complex track, except for the Taxon dataset, which lacks a reference alignment. Instance-based metrics, on the other hand, can only evaluate five of the seven datasets and fully analyze only two (populated_conference and Taxon) due to the strict requirements of having both CQAs and instances. For datasets like populated GeoLink, Hydrography, and populated Enslaved, only precision evaluation is feasible. However, this evaluation can complement the proposed one when no reference alignment is present in the dataset.

4 Related Work

In complex ontology matching (the reader can refer to [12] for a survey), manual evaluation remains a technique used to assess matchers' performance due to the intricate nature of the correspondences involved [7, 6, 14]. Unlike simple matching, where automated metrics like precision and recall can often provide reliable assessments, complex matching requires deeper semantic understanding and contextual interpretation that automated tools have difficulties achieving. Human experts are typically involved in evaluating the alignments' correctness, completeness, and semantic coherence, particularly in scenarios where gold standards are unavailable or incomplete. This manual process is time-consuming and subject to potential biases or inconsistencies.

To enable automatic evaluation of complex ontology matching, one of the commonly used metrics is relaxed precision and recall [2]. These metrics aim to provide flexibility by allowing partial matches between complex correspondences rather than requiring exact equivalence. However, a significant limitation is that, while being a general metric that can be extended, they do not define metrics to measure the similarity between the subgraphs or structures involved in complex correspondences. Consequently, relaxed precision and recall without the

proper graph similarity score may fail to capture the full structural and semantic fidelity of complex correspondences, limiting their effectiveness in assessing overall alignment quality.

Other metrics for evaluating complex ontology alignments rely on set-based computations, focusing on the overlap of common entities between the corresponding elements as described in [9]. A common metric applied to the complex matching case based on this principle is the measure of the number of correct entities present in the correspondence, resembling the Jaccard similarity metric. While these metrics are straightforward and computationally efficient, they have limitations in representativity since they don't consider the structural relationships or semantic dependencies between the entities within the subgraphs defined by the correspondences. This omission means that approaches capable of correctly identifying those aggregation relations won't have higher performance than just identifying the composing entities.

Another approach to evaluating complex ontology alignments involves metrics based on Competency Questions for Alignment (CQAs) and instances [13]. CQAs are SPARQL queries written to describe the user needs in terms of alignment, and are often provided as input to the matchers in some datasets. In that proposed evaluator, the quality of the alignment is assessed by comparing the similarity between the CQAs and the rewritten alignments as SPARQL queries. However, this method faces significant challenges, as CQAs are user-defined and require manual creation, making the process time-intensive and dependent on human expertise. Additionally, some metrics evaluate the alignment by measuring the number of common instances retrieved by the rewritten queries. While this can provide valuable insights, it has critical limitations: not all ontologies contain instances, and even when instances exist, not all entities have associated instances. As a result, these metrics fail to comprehensively evaluate all matched concepts, leaving gaps in the assessment of alignment quality, since it may not cover the whole ontology.

5 Conclusion

This paper introduced a novel metric to evaluate complex ontology alignments that consider the structure of matched entities. While some metrics rely on the use of common instances (A-Box data), such an approach is not always feasible, as ontologies may not be exhaustively populated or may lack an A-Box altogether. Our metric provides an alternative in these scenarios. In the OAEI complex track, most datasets include reference alignments, but these are often underexploited — either partially or evaluated manually. Unlike existing approaches, the proposed metric accounts for the structural and semantic relations of (m:n) correspondences by using a tree similarity computation and ensuring adherence to key evaluation properties for alignments written in EDOAL, such as identity, order invariance, and sensitivity to correspondence modifications. The empirical analysis demonstrated the metric's capability to deliver insights

into alignment quality compared to traditional and instance-based evaluation methods.

The proposed metric outperforms alternatives in scenarios emphasizing structural evaluation, enabling a more accurate distinction between simple and complex correspondences. It also addresses limitations in other methods, such as their inability to comprehensively evaluate alignments that involve structural components or their reliance on dataset-specific characteristics like instance availability or predefined competency questions. This structural evaluation requires that matchers built for the complex task produce the correct combination structure for the entities. Now with LLMs that combination can be better produced, and having suitable metrics can help the complex evaluation task follow the advancements of techniques in this field. Also, a new evaluation technique can bring new perspectives to the Complex track in OAEI evaluation and help bring more participants.

However, challenges remain. The dependency on high-quality reference alignments can limit applicability in contexts where such alignments are unavailable or prone to errors. Future work will explore approaches to mitigate these dependencies, such as developing probabilistic models to assess alignment quality in the absence of complete gold standards. Furthermore, refinement of the metric to accommodate evolving ontology formats and emerging alignment needs will ensure its continued relevance in advancing ontology matching research.

References

1. Amini, R., Norouzi, S.S., Hitzler, P., Amini, R.: Towards complex ontology alignment using large language models. CoRR **abs/2404.10329** (2024). https://doi.org/10.48550/ARXIV.2404.10329, https://doi.org/10.48550/arXiv.2404.10329
2. Ehrig, M., Euzenat, J.: Relaxed precision and recall for ontology matching. In: Ashpole, B., Ehrig, M., Euzenat, J., Stuckenschmidt, H. (eds.) Integrating Ontologies '05, Proceedings of the K-CAP 2005 Workshop on Integrating Ontologies, Banff, Canada, October 2, 2005. CEUR Workshop Proceedings, vol. 156. CEUR-WS.org (2005), https://ceur-ws.org/Vol-156/paper5.pdf
3. Euzenat, J., David, J., Atencia, M.: Edoal: Expressive and declarative ontology alignment language. URL: http://alignapi. gforge. inria. fr/edoal. html (visited on 2/12/2024) (2015)
4. Jouhet, V., Mougin, F., Bréchat, B., Thiessard, F.: Building a model for disease classification integration in oncology, an approach based on the national cancer institute thesaurus. Journal of Biomedical Semantics **8**(1), 6:1–6:12 (2017). https://doi.org/10.1186/s13326-017-0114-4
5. Nurmikko-Fuller, T., Page, K.R., Willcox, P., Jett, J., Maden, C., Cole, T.W., Fallaw, C., Senseney, M., Downie, J.S.: Building complex research collections in digital libraries: A survey of ontology implications. In: Proceedings of the 15th ACM/IEEE-CE Joint Conference on Digital Libraries. pp. 169–172. ACM (2015). https://doi.org/10.1145/2756406.2756944
6. Parundekar, R., Knoblock, C.A., Ambite, J.L.: Discovering concept coverings in ontologies of linked data sources. In: Cudré-Mauroux, P., Heflin, J., Sirin, E., Tudorache, T., Euzenat, J., Hauswirth, M., Parreira, J.X., Hendler, J., Schreiber,

G., Bernstein, A., Blomqvist, E. (eds.) The Semantic Web - ISWC 2012 - 11th International Semantic Web Conference, Boston, MA, USA, November 11-15, 2012, Proceedings, Part I. Lecture Notes in Computer Science, vol. 7649, pp. 427–443. Springer (2012). https://doi.org/10.1007/978-3-642-35176-1_27, https://doi.org/10.1007/978-3-642-35176-1_27

7. Ritze, D., Völker, J., Meilicke, C., Sváb-Zamazal, O.: Linguistic analysis for complex ontology matching. In: Shvaiko, P., Euzenat, J., Giunchiglia, F., Stuckenschmidt, H., Mao, M., Cruz, I.F. (eds.) Proceedings of the 5th International Workshop on Ontology Matching (OM-2010), Shanghai, China, November 7, 2010. CEUR Workshop Proceedings, vol. 689. CEUR-WS.org (2010), https://ceur-ws.org/Vol-689/om2010_Tpaper1.pdf
8. Silva, M.C., Faria, D., Pesquita, C.: Complex multi-ontology alignment through geometric operations on language embeddings. In: Endriss, U., Melo, F.S., Bach, K., Diz, A.J.B., Alonso-Moral, J.M., Barro, S., Heintz, F. (eds.) ECAI 2024 - 27th European Conference on Artificial Intelligence, 19-24 October 2024, Santiago de Compostela, Spain - Including 13th Conference on Prestigious Applications of Intelligent Systems (PAIS 2024). Frontiers in Artificial Intelligence and Applications, vol. 392, pp. 1333–1340. IOS Press (2024). https://doi.org/10.3233/FAIA240632, https://doi.org/10.3233/FAIA240632
9. Thiéblin, É.: Automatic Generation of Complex Ontology Alignments. (Génération automatique d'alignements complexes d'ontologies). Ph.D. thesis, Paul Sabatier University, Toulouse, France (2019), https://tel.archives-ouvertes.fr/tel-02735724
10. Thiéblin, E., Amarger, F., Hernandez, N., Roussey, C., Trojahn, C.: Cross-querying LOD datasets using complex alignments: An application to agronomic taxa. In: Metadata and Semantic Research - 11th Inter. Conference, MTSR. vol. 755, pp. 25–37 (2017). https://doi.org/10.1007/978-3-319-70863-8_3
11. Thiéblin, É., Cheatham, M., Trojahn, C., Zamazal, O.: A consensual dataset for complex ontology matching evaluation. Knowl. Eng. Rev. **35**, e34 (2020). https://doi.org/10.1017/S0269888920000247, https://doi.org/10.1017/S0269888920000247
12. Thiéblin, É., Haemmerlé, O., Hernandez, N., Trojahn, C.: Survey on complex ontology matching. Semantic Web **11**(4), 689–727 (2020). https://doi.org/10.3233/SW-190366, https://doi.org/10.3233/SW-190366
13. Thiéblin, É., Haemmerlé, O., Trojahn, C.: Automatic evaluation of complex alignments: An instance-based approach. Semantic Web **12**(5), 767–787 (2021). https://doi.org/10.3233/SW-210437, https://doi.org/10.3233/SW-210437
14. Walshe, B., Brennan, R., O'Sullivan, D.: Bayes-recce: A bayesian model for detecting restriction class correspondences in linked open data knowledge bases. Int. J. Semantic Web Inf. Syst. **12**(2), 25–52 (2016). https://doi.org/10.4018/IJSWIS.2016040102, https://doi.org/10.4018/IJSWIS.2016040102
15. Zhang, K., Shasha, D.E.: Simple fast algorithms for the editing distance between trees and related problems. SIAM J. Comput. **18**(6), 1245–1262 (1989). https://doi.org/10.1137/0218082, https://doi.org/10.1137/0218082
16. Zhang, K., Statman, R., Shasha, D.E.: On the editing distance between unordered labeled trees. Inf. Process. Lett. **42**(3), 133–139 (1992). https://doi.org/10.1016/0020-0190(92)90136-J, https://doi.org/10.1016/0020-0190(92)90136-J