Towards Association Rule-Based Complex Ontology Alignment

Lu Zhou[1], Michelle Cheatham[2], and Pascal Hitzler[1]

[1] DaSe Lab, Kansas State University, Manhattan KS 66506, USA
{luzhou, hitzler}@ksu.edu
[2] Wright State University, Dayton OH 45435, USA
michelle.cheatham@wright.edu

Abstract. Ontology alignment has been studied for over a decade, and over that time many alignment systems have been developed by researchers in order to find simple 1-to-1 equivalence alignments between ontologies. However, finding complex alignments, i.e., alignments that are not simple class or property equivalences, is a topic largely unexplored but with growing significance. Currently, establishing a complex alignment requires domain experts to work together to manually generate the alignment, which is extremely time-consuming and labor-intensive. In this paper, we propose an automated method based on association rule mining to detect not only simple alignments, but also more complex alignments between ontologies. Our algorithm can also be used in a semi-automated fashion to effectively assist users in finding potential complex alignments which they can then validate or edit. In addition, we evaluate the performance of our algorithm on the complex alignment benchmark of the Ontology Alignment Evaluation Initiative (OAEI).

1 Introduction

Ontology alignment is an important step in enabling computers to query and reason across the many linked datasets on the semantic web. This is a difficult challenge because the ontologies underlying different linked datasets can vary in terms of subject area coverage, level of abstraction, ontology modeling philosophy, and even language. Due to the importance and difficulty of the ontology alignment problem, it has been an active area of research for over a decade [21].

Ideally, alignment systems should be able to uncover any entity relationship across two ontologies that can exist within a single ontology. Such relationships have a wide range of complexity, from simple 1-to-1 equivalence, such as a Person in one ontology being equivalent to a Human in another ontology, to arbitrary m-to-n complex relationships, such as a Professor with a hasRank property value of "Assistant" in one ontology being a subclass of the union of the Faculty and TenureTrack classes in another. Unfortunately, the majority of the research activities in the field of ontology alignment remains focused on the simplest end of this scale – finding 1-to-1 equivalence alignments between ontologies. Indeed, identifying arbitrarily complex alignment is known to be significantly harder than finding 1-to-1 equivalences. In the latter case, a naive approach can compare every entity from the source ontology against every entity in the target ontology, which is feasible for small- and medium-sized ontologies. However, a complex alignment can potentially involve many entities from both ontologies, so pair-wise comparison is insufficient, and the search space become very large even for small ontologies. It is indeed very difficult for either a human expert or an automated system to evaluate all possible combinations [2, 19].

In this paper, we propose a complex alignment algorithm based on association rule mining. Our algorithm automatically discovers potential complex correspondences which can then be presented to human experts in order to effectively generate complex alignment between two ontologies with populated common instance data. We evaluate the performance of our system on one of the benchmarks from the complex alignment track of the OAEI 2018,[3] the GeoLink benchmark, which contains around 74k instances from real-world datasets. Significant instance data, which is required for the association rule mining approach, is not available for the remaining benchmarks.[4] The main contributions of this paper are the following:

– The association rule-based algorithm automatically detects not only 1-to-1 equivalences, but also more complex alignment between two ontologies.
– A detailed analysis of the results provides a good understanding of the efficacy of this approach and identifies further directions for advancement.

There is a side contribution when we analyze the results, which is that our algorithm shows that shared instance data between two ontologies can be a good resource to improve the performance of ontology alignment.

The rest of the paper is organized as follows. Section 2 discusses related work in ontology alignment using association rule mining and instance data and complex ontology alignment, including existing alignment algorithms and relevant benchmarks. Section 3 gives background on the FP-growth association rule mining algorithm. Section 4 illustrates the association rule-based alignment algorithm in detail, along with the alignment patterns used to generate the alignment between ontologies. The analysis of the performance of the system is discussed in Section 5. Section 6 concludes with a discussion of potential future work in this area.

2 Related Work

Association rule mining has already been used for finding 1:1 simple alignments. AROMA [4] is a hybrid, extensional and asymmetric ontology alignment method that makes use of association rules and a statistical measure. It relies on the idea that "An entity A will be more specific than or equivalent to an entity B if the vocabulary used to describe A and its instances tends to be included in that of B and its instances." In addition, association rule mining is also used in discovering rules in ontological knowledge bases [10] and logical linked data compression [15].

[3] http://oaei.ontologymatching.org/2018/complex/index.html
[4] It might be available for OAEI 2019.

There are also some instance-based ontology alignment systems that utilize Abox information to generate 1:1 simple alignments between ontologies. GLUE [6] uses joint probability distributions to describe the similarity of concepts in two ontologies. For example, $p(A, B)$ is the probability that an instance in the domain belongs to both concept A and concept B. And then, if the instances of concept A and concept B are in isolation, GLUE uses the instances of A to learn a classifier for A, and then classifies instances of B according to that classifier, and vice-versa. FCA_MERGE also utilizes common instances between ontologies [22]. FCA_MERGE extracts instances from a given set of domain-specific text documents by applying nature language processing techniques. Based on the extracted instances, FCA_MERGE applies mathematical techniques to derive a lattice of concepts as a structural result of FCA_MERGE. More instance-based alignment systems have been discussed in the survey [26].

There are some related studies on creating algorithms to find complex alignment between ontologies. Early work on generating complex alignment is [19, 20]. Therein, three complex alignment patterns were described, which are Class by Attribute Type (CAT), Class by Attribute Value (CAV), and Property Chain (PC). Based on these patterns, the authors generated complex alignments on the Conference and Benchmark datasets from the OAEI. [13] identified complex alignments by defining knowledge rules and using a probabilistic framework to integrate a knowledge-based strategy with standard terminology-based and structure-based strategies. More recent related work is currently being undertaken by Thieblin et al. [24]. They propose a complex alignment approach that relies on the notion of Competency Question for Alignment (CQA). The approach translates a CQA into a SPARQL query and extracts a set of instance data from the source ontology. Then the matching is performed by finding the lexically similar surroundings between the set of instance data and the instances in the target ontology. This approach resulted in the CANARD system [23]. However, the current version of the system is limited to finding complex correspondences that only involve classes. More complex correspondences containing properties are still not taken into account [23]. Another alignment system that works on the detection of the complex alignment is the complex version of AgreementMakerLight (AMLC) [9]. This system focuses on the complex Conference benchmark to find alignments that follow the CAT and CAV patterns.

In OAEI 2018, the first version of the complex alignment track [25] opened new perspectives in the field of ontology matching. It comprised four different benchmarks containing complex relations. However, the results from the first year were rather poor. Only 2 out of 15 systems, AMLC and CANARD, were able to generate any correct complex correspondences on the complex Conference and Taxon benchmarks, and the correct number of mappings found was quite limited. The very limited performance of the two systems of course shows avenues for improvement in the future. More details of evaluations and results can be accessed on the OAEI 2018 website.[5]

[5] http://oaei.ontologymatching.org/2018/complex/index.html

Our algorithm differs from the above methods in several aspects. First, [9], [13], and [19] focus on computing lexical or terminological similarity to decide on complex alignments, while our system takes advantage of instance data to generate association rules between ontologies. While the CANARD system also relies on the instance data, we use it in completely different ways. In addition, the current version of CANARD is limited to finding complex correspondences that involve only classes, while our algorithm does not have this limitation. Second, our evaluation of results is more detailed, in order to provide insight into how to improve the performance of complex alignment algorithms. Specifically, we break the evaluation process down into two subtasks: entity identification and relationship identification. We utilize a variation of traditional evaluation metrics called relaxed precision, recall, and f-measure [7] to present the final results of the full complex alignment.

3 Background

In order to help the reader understand how we apply association rule mining and the FP-growth algorithm on the ontology alignment task, we introduce here some concepts that we frequently mention in the rest of the paper.

Association Rule Mining. Our alignment system mainly depends on a data mining algorithm called association rule mining, which is a rule-based machine learning method for discovering interesting relations between variables in large databases [17]. Over the years, association rule mining has played an important role in many data mining tasks, such as market basket analysis, web usage mining, and bioinformatics. Many algorithms for generating association rules have been proposed, like Apriori [1] and FP-growth algorithm [11]. In this paper, we use FP-growth to generate association rules between ontologies, since the FP-growth algorithm has been proven superior to other algorithms [11] and will improve the algorithm in terms of run-time.

Transaction Database. Let $I = \{i_1, i_2, \dots, i_n\}$ be a set of distinct attributes called items. Let $D = \{t_1, t_2, \dots, t_m\}$ be a set of transactions where each transaction in D has a unique transaction ID and contains a subset of the items in I. Table 1 shows a list of transactions corresponding to a list of triples. The data in an ontology can be displayed as a set of triples, each consisting of subject, predicate, and object. Here, subjects represent the identifiers and the set of corresponding properties with the objects represent transactions, which are separated by the symbol "|". I.e., a transaction is a set $T = (s, Z)$ such that s is a subject, and each member of Z is a pair (p, o) of a property and an object such that (s, p, o) is a triple.

FP-growth. The FP stands for frequent pattern. The FP-growth algorithm is run on the transaction database in order to determine which combinations of items co-occur frequently. The algorithm first counts the number of occurrences of all individual items in the database. Next, it builds an FP-tree structure by inserting these instances. Items in each instance are sorted by descending order of their frequency in the dataset, so that the tree can be processed quickly. Items

Table 1. Triples and Corresponding Transactions

s_1	p_1	o_1
s_1	p_2	o_2
s_1	p_4	o_4
s_2	p_1	o_1
s_2	p_2	o_2
s_2	p_3	o_3
s_2	p_4	o_4
s_3	p_1	o_1
s_3	p_2	o_2

TID	**Itemsets**
s_1	$p_1\|o_1$, $p_2\|o_2$, $p_4\|o_4$
s_2	$p_1\|o_1$, $p_2\|o_2$, $p_3\|o_3$, $p_4\|o_4$
s_3	$p_1\|o_1$, $p_2\|o_2$

Table 2. Examples of Association Rules

Antecedent	**Consequent**
$p_4\|o_4$, $p_1\|o_1$	$p_2\|o_2$
$p_2\|o_2$	$p_1\|o_1$
$p_4\|o_4$	$p_1\|o_1$

in each instance that do not meet the predefined thresholds, such as minimum support and minimum confidence (see below for these terms), are discarded. Once all large itemsets have been found, the association rule creation begins.

Association Rule. Every association rule is composed of two sides. The left-hand-side is called the antecedent, and the right-hand-side is the consequent. These rules indicate that whenever the antecedent is present, the consequent is likely to be as well. Table 2 shows some examples of association rules generated from the transaction database in Table 1.

Support. Support indicates how frequently an itemset appears in the dataset. The FP-growth algorithm finds the frequent itemsets from the dataset based on the minimum support threshold. In our alignment system, the minimum support value is examined and set to 0.001 to guarantee the best performance.

Confidence. Confidence is an indication of how often an association rule has been found to be true, i.e. how often the presence of the antecedent is associated with the presence of the consequent. The minimum confidence can be tuned to find relatively accurate rules. In this paper, we use the minimum confidence of 0.3 as default value. And we tune the value to 1 when we mine the association rules that may contain complex relations, because our algorithm would focus on precision-oriented results.

Lift. Lift is the ratio of the observed support to that expected if the antecedent and consequent were independent. If the lift is greater than 1, it means that the two items are dependent on one another, which indicates that the association rule useful. In our approach, lift is used to choose between otherwise equal options when detecting simple mappings. When the confidence values of two association rules are the same, the one with higher lift value is selected as the basis for the mapping.

4 Association Rule-Based Alignment Algorithm

In this section, we introduce the proposed ontology alignment algorithm based on association rule mining in detail. Figure 1 illustrates the overview of our proposed algorithm.

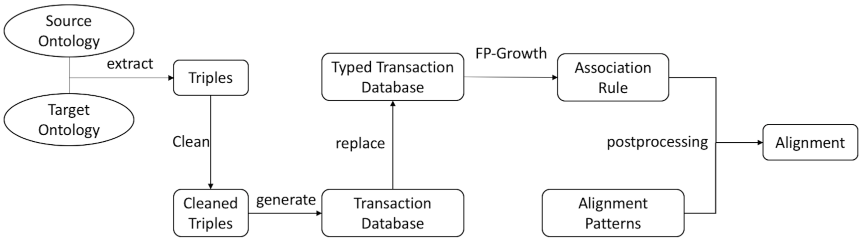

Fig. 1. Overview of The Proposed Alignment Algorithm

4.1 Data Preparation

We first extract all triples ⟨Subject, Predicate, Object⟩ from the source and target ontologies. Each item in a triple is expressed as a web URI. After collecting all of the triples, we prepare the data as follows: we only keep the triples that contain at least one entity under the source or the target ontology namespace and also the triples that contain rdf:type information, since our algorithm relies on this information. After this, there are still some triples containing less useful information for association rule mining, which follow this format: x rdf:type owl:NamedIndividual. This triple is not very informative except stating the subject x is an individual. But, it frequently occurs in the dataset and may lead to noises when applying the FP-growth algorithm, since the frequency of occurrence impacts the results of FP-growth. So, we filter out such noise from the dataset as well.

After this filtering process, we generate the transaction database for the FP-growth algorithm based on all of the remaining triples. The subjects serve as the transaction IDs, and the predicates with the objects separated by the symbol "|" are the items for each transaction. Then we replace the object in the triples with its rdf:type,[6] because we focus on generating schema-level (rather than instance-level) mapping rules between two ontologies, and the type information of the object is more meaningful than the original URI. If an object in a triple has rdf:type of a class in the ontology, we replace the URI of the object with its class. If the object is a data value, the URI of the object is replaced with the datatype. If the object already is a class in the ontology, it remains unchanged. Tables 3 and 4 show some examples of the conversion.

4.2 Association Rule and Alignment Generation

We run the FP-growth algorithm on the transaction database and generate a set of association rules. Since we are trying to find the mappings *between* two ontologies, we focus on mining the rules whose antecedent only contains entities

[6] Our evaluation data has only single type. If there are multiple types of the object, it can also combine the subject and predicate as additional information to determine the correct type, or keep both types as two triples.

Table 3. Original Transaction Database

TID	Itemsets
x_1	gbo:hasAward|y$_1$, gmo:fundedBy|y$_2$
x_2	gbo:hasFullName|y$_3$, gmo:hasPersonName|y$_4$
x_3	rdf:type|gbo:Cruise, rdf:type|gmo:Cruise

Table 4. Typed Transaction Database

TID	Itemsets
x_1	gbo:hasAward|gbo:Award, gmo:fundedBy|gmo:FundingAward
x_2	gbo:hasFullName|xsd:string, gmo:hasPersonName|gmo:PersonName
x_3	rdf:type|gbo:Cruise, rdf:type|gmo:Cruise

from the source ontology and whose consequent only contains entities from the target ontology. The association rules tell us which source entities are related to which target entities, but they do not give us information on *how* those entities are related. In order to determine this, we analyze the output of the association rule mining step in light of the common alignment patterns introduced in [19, 27]. In the following, we introduce how we leverage these alignment patterns to filter the association rules and generate the corresponding alignment. The following examples that we use in this paper are from the GeoLink benchmark [27]. gbo: is the prefix of the namespace of the GeoLink Base Ontology (GBO), and gmo: is the prefix of the namespace of the GeoLink Modular Ontology (GMO). The alignment between the two ontologies contains both simple and complex correspondences. To deal with the redundancy of generated association rules, we always keep the simpler rule as the result. For example, there are two association rules generated by our system. Cruise in the GBO is equivalent to the domain of fundedBy with it range of FundingAward in the GMO. And Cruise in the GBO is also equivalent to Cruise in the GMO, which is the domain of fundedBy. Therefore, the two mapping rules are semantically equivalent. And we only keep the second rule which is the simpler one as our result.

Simple Alignment. Simple alignment is a set of simple correspondences that refer to basic 1-to-1 simple mappings between two ontologies, in which the entities involved may be either classes or properties.

1:1 Class Alignment. The first pattern is simple 1-to-1 class relationships. Classes C_1 and C_2 are from ontology O_1 and ontology O_2, respectively. So, we target the association rules with the following format:

Association Rule format: rdf:type|C_1 → rdf:type|C_2
Example: rdf:type|gbo:Award → rdf:type|gmo:FundingAward
Generated Alignment: gbo:Award(x) → gmo:FundingAward(x)

The left and right hand side of the arrow represent the antecedent and consequent in the association rules, respectively. In the example, the association rule implies that if an individual x has rdf:type of gbo:Award, then x also has rdf:type of gmo:FundingAward. This means that gbo:Award is a subclass of gmo:FundingAward. If there is another association rule containing the reverse information, which means that gmo:FundingAward is also a subclass of gbo:Award then we can generate an alignment based on the two association rules stating that gbo:Award is equivalent to gmo:FundingAward. This method of choosing between subsumption and equivalence relationships is used for all of the following types of correspondences as well.

1:1 Property Alignment. This pattern captures simple 1-to-1 property mappings. The property can be either an object property or a data property.

(1) Object Property Alignment. Since we have the information of the type of the object in the association rule, we can use the type information to filter the mapping candidates. When we align two object properties, the range types of the properties are usually either equivalent to each other or compatible (because they are in a subclass or superclass relationship). In this paper, our algorithm is precision-oriented. Therefore, we require the object properties in the two ontologies to have equivalent (rather than compatible) ranges in order to be considered equivalent. Range equivalence is determined through the results of the simple class alignment introduced above. Object Property op_1 with its range type t_1 and object property op_2 with its range type t_2 are from ontology O_1 and ontology O_2, respectively. In order to find this alignment, we select the association rules with the following format:

Association Rule format: $op_1|t_1 \rightarrow op_2|t_2$
Example: $\mathsf{gbo:hasAward|gbo:Award} \rightarrow \mathsf{gmo:fundedBy|gmo:FundingAward}$
Generated Alignment: $\mathsf{gbo:hasAward(x,y)} \rightarrow \mathsf{gmo:fundedBy(x,y)}$

We know from the results of the simple class alignment that gbo:Award is equivalent to gmo:FundingAward. This association rule says that gbo:hasAward is subsumed by gmo:fundedBy. If there is another association rule containing the reverse relationship, we can generate the mapping that gbo:hasAward is equivalent to gmo:fundedBy.

(2) Data Property Alignment. Similar to aligning object properties, when aligning two data properties, the range values of the two properties should be of a compatible datatype. In this paper, we only investigate equivalent datatypes. Data Property dp_1 with its range value t_1 and property dp_2 with its range value t_2 are from ontology O_1 and ontology O_2, respectively.

Association Rule format: $dp_1|t_1 \rightarrow dp_2|t_2$
Example:
$\mathsf{gbo:hasIdentifierValue|xsd:string} \rightarrow \mathsf{gmo:hasIdentifierValue|xsd:string}$
Generated Alignment:
$\mathsf{gbo:hasIdentifierValue(x,y)} \rightarrow \mathsf{gmo:hasIdentifierValue(x,y)}$

(3) Data/Object to Object/Data Property Alignment. It is possible that two ontologists may model the same property differently – e.g., there is an example in the OAEI GeoLink complex alignment benchmark [27]. The entity

hasIdentifierScheme is modeled as an object property in the GBO with a range of class IdentifierScheme. But, this entity is modeled as a data property in the GMO with a range of the string datatype. In this case, we calculate the Levenshtein string similarity between the labels of the two properties and keep the pairs within a predefined threshold (0.9 is examined to get the best performance). The association rule should have the following format:

Association Rule format: $op_1/dp_1|t_1 \rightarrow dp_2/op_2|t_2$
Example:
$\mathsf{gbo{:}hasIdentifierScheme|gbo{:}IdentifierScheme} \rightarrow$
$\mathsf{gmo{:}hasIdentifierScheme|xsd{:}string}$
Generated Alignment:
$\mathsf{gbo{:}hasIdentifierScheme(x, y)} \rightarrow \mathsf{gmo{:}hasIdentifierScheme(x, y)}$

Complex Alignment. Complex alignment is a set of Complex correspondences that refer to more complex patterns, such as 1-to-n equivalence, 1-to-n subsumption, m-to-n equivalence, m-to-n subsumption, and m-to-n arbitrary relationship.

1:n Class Alignment. This type of pattern was first introduced in [19]. It contains two different patterns: the Class by Attribute Type pattern (CAT) and the Class by Attribute Value pattern (CAV). In addition, [27] introduced another pattern called Class Typecasting.

(4) Class by Attribute Type. This pattern states that a class in the source ontology is in some relationship to a complex construction in the target ontology. This complex construction may comprise an object property and its range type. Class C_1 is from ontology O_1, and object property op_1 and its range type t_1 are from ontology O_2.

Association Rule format: $\mathsf{rdf{:}type}|C_1 \rightarrow \mathsf{op_1|t_1}$
Example: $\mathsf{rdf{:}type|gbo{:}PortCall} \rightarrow \mathsf{gmo{:}atPort|gmo{:}Place}$
Generated Alignment: $\mathsf{gbo{:}PortCall(x)} \rightarrow \mathsf{gmo{:}atPort(x, y)} \wedge \mathsf{gmo{:}Place(y)}$

In this example, this association rule implies that if the subject x is an individual of class gbo:PortCall, then x is subsumed by the domain of gmo:atPort with the range type of gmo:Place. The equivalence relationship can be generated by combining another association rule holding the reverse information.

(5) Class by Attribute Value. This pattern is similar to the previous one. It just replaces the object property with a data property. Class C_1 is from ontology O_1, and data property dp_1 and its datatype of the range value t_1 are from ontology O_2.

Association Rule format: $\mathsf{rdf{:}type}|C_1 \rightarrow \mathsf{dp_1|t_1}$
Example: $\mathsf{rdf{:}type|gbo{:}Identifier} \rightarrow \mathsf{gmo{:}hasIdentifierScheme|xsd{:}string}$
Generated Alignment: $\mathsf{gbo{:}Identifier(x)} \rightarrow \mathsf{gmo{:}hasIdentifierScheme(x, y)}$

(6) Class Typecasting. This pattern indicates that an individual x of type C_1 in one ontology O_1 is cast into a subclass of C_2 in the other ontology O_2.

Association Rule format: $\mathsf{rdf{:}type}|C_1 \rightarrow \mathsf{rdfs{:}subClassOf}|C_2$
Example: $\mathsf{gbo{:}PlaceType} \rightarrow \mathsf{rdfs{:}subClassOf|gmo{:}Place}$
Generated Alignment: $\mathsf{gbo{:}PlaceType} \rightarrow \mathsf{rdfs{:}subClassOf(x, gmo{:}Place)}$

1:n Property Alignment This pattern represents a Property Typecasting relationship that is defined in [27].

(7) 1:n Property Typecasting. This pattern is similar in spirit to the Class Typecasting patterns mentioned above. However, in this case, a property from one ontology is cast into a class assignment statement in the other ontology.

Association Rule format: $p_1|t_1 \rightarrow \mathsf{rdf{:}type}|C_2$
Example: $\mathsf{gbo{:}hasPlaceType}|\mathsf{gbo{:}PlaceType} \rightarrow \mathsf{rdf{:}type}|\mathsf{gmo{:}Place}$
Generated Alignment:
$\mathsf{gbo{:}hasPlaceType}(\mathsf{x},\mathsf{y}) \wedge \mathsf{gbo{:}PlaceType}(\mathsf{y}) \rightarrow \mathsf{gmo{:}Place}(\mathsf{x})$

m:n Complex Alignment. This group contains the most complex correspondences.

(8) m:n Property Chain. This pattern applies, for example, when a property, together with type restrictions on one or both of its fillers, in one ontology, has been used to "flatten" the structure of the other ontology by short-cutting a property chain in that ontology. The pattern also ensures that the types of the property fillers involved in the property chain are typed appropriately in the other ontology. The class C_1 and property r_1 with its range restriction t_1 are from ontology O_1, and classes B_i and properties p_i with its range restriction d_i are from ontology O_2.

Association Rule format:

$$\mathsf{rdf{:}type}|\mathsf{C_1}, \mathsf{r_1}|\mathsf{t_1} \rightarrow \mathsf{rdf{:}type}|\mathsf{B_1}, \mathsf{p_1}|\mathsf{d_1}, \ldots, \mathsf{rdf{:}type}|\mathsf{B_i}, \mathsf{p_i}|\mathsf{d_i}$$

Example:

$$\begin{aligned}&\mathsf{gbo{:}Award}, \mathsf{gbo{:}hasSponsor}|\mathsf{gbo{:}Organization}\\ &\qquad \rightarrow \mathsf{rdf{:}type}|\mathsf{gmo{:}FundingAward},\\ &\qquad\quad \mathsf{gmo{:}providesAgentRole}|\mathsf{gmo{:}SponsorRole},\\ &\qquad\quad \mathsf{gmo{:}performedBy}|\mathsf{gmo{:}Organization}\end{aligned}$$

Generated Alignment:

$$\begin{aligned}&\mathsf{gbo{:}Award}(\mathsf{x}) \wedge \mathsf{gbo{:}hasSponsor}(\mathsf{x},\mathsf{z}) \wedge \mathsf{gbo{:}Organization}(\mathsf{z})\\ &\qquad \rightarrow \mathsf{rdf{:}type}|\mathsf{gmo{:}FundingAward}(\mathsf{x})\wedge\\ &\qquad\quad \mathsf{gmo{:}providesAgentRole}(\mathsf{x},\mathsf{y}) \wedge \mathsf{gmo{:}SponsorRole}(\mathsf{y})\wedge\\ &\qquad\quad \mathsf{gmo{:}performedBy}(\mathsf{y},\mathsf{z}) \wedge \mathsf{gmo{:}Organization}(\mathsf{z})\end{aligned}$$

In this example, the association rule implies that in the GBO, the property gbo:hasSponsor with the domain type of gbo:Award and the range type of gbo:Organization has been used to "flatten" the complex structure in the GMO by short-cutting a property chain. Note that in this pattern, C_1 and any of the B_i may be omitted (in which case they are essentially $\top$).

5 Evaluation

In this section, we show the experimental results of our proposed alignment algorithm on the OAEI GeoLink benchmark and analyze the results in detail. The GeoLink benchmark [27] is composed of two ontologies in the geosciences domain. These two ontologies are both populated with 100% shared instance data collected from the real-world GeoLink knowledge base [3], in order to help the evaluation of alignment algorithms depending on instance data.[7] The subset used for this study contains around 74k triples, which is suitable for applying association rule mining.

We originally planned to compare the performance of our system against pattern based system in [19], CANARD, and AMLC. However, the GeoLink benchmark is a property-oriented dataset which involves many object or data properties in the complex correspondences. As we discussed in Section 2, CANARD is currently limited to finding complex mappings that only involve classes. Even though pattern based system in [19] can generate property-based complex correspondences, like property chains, there are several rules that the system follows that largely limit its results, and it ends without finding any complex alignment on the GeoLink ontology pair. AMLC currently only works for the complex Conference benchmark [2, 9]. Therefore, there are no complex alignment systems against which we could compare the performance of our system. So in this paper we are limited to reporting the performance of our system against the reference alignment when it comes to the identification of complex alignment. Performance on the identification of simple alignment is compared against that of systems that participated in the OAEI 2018.

Because the systems we compare against are only capable of identifying simple correspondences, we present the results on the simple and complex portions of the overall alignment separately.[8] For simple correspondences, we use the traditional precision, recall and F-measure metrics, in order to compare against other simple alignment systems. However, in order to provide more insight into the underlying nature of the performance on complex correspondences, we take a slightly different approach. Semantic precision and recall, which compare correspondences based on their semantic meaning rather than their syntactic representation [8]. This is done by applying a reasoner to determine when one mapping is logically equivalent to another. Even though the semantic approaches solve an important problem for evaluating alignments with complex correspondences, they still have several limitations. One is that the reasoning takes a significant amount of time, particularly for large ontologies. Furthermore, such reasoning is not possible if the merged ontology is not in OWL DL. The GeoLink benchmark is one example of this case, since there are many correspondences involving an object property on one side and a data property on another side, which is not permissible in OWL DL. Instead, we utilize relaxed precision and recall [7]. More specifically, a correspondence consists of two aspects: the entities involved, and the relationship between them (e.g. equivalence, subsumption, disjunction). In order to assess performance on both of these aspects, we evaluate them separately. This roughly corresponds to the first and second subtasks described for some of the test sets within the complex track of the OAEI.[9] However, the types

[7] https://doi.org/10.6084/m9.figshare.5907172

[8] We are aware that this may not be the most general way to evaluate complex alignments, but the community does not yet have any guidelines or tangible results which could be used. And solving the evaluation problem is out of scope of this paper.

[9] http://oaei.ontologymatching.org/2018/complex/index.html#hydrography

Table 5. The Performance Comparison of Matchers on the Simple Alignment

Matcher	# of 1:1 Class Equiv.	# of 1:1 Class Subsum.	# of 1:1 Property Equiv.	Precision	Recall	F-measure
Reference Alignment	10	2	7	-	-	-
Our Results	10	0	5	**0.94**	**0.79**	**0.86**
CANARD [18]	9	0	3	0.67	0.63	0.64
DOME [12]	9	0	4	0.41	0.68	0.51
LogMap [14]	9	0	1	0.77	0.53	0.63
LogMapKG [14]	9	0	1	0.77	0.53	0.63
LogMapLt [14]	9	0	5	0.63	0.73	0.68
POMAP++ [16]	9	0	0	0.89	0.47	0.62
XMap [5]	9	0	0	0.39	0.47	0.43

of relationships we consider are limited to equivalence and subsumption rather than the arbitrary OWL constructs considered there.

5.1 Simple Alignment Evaluation

In the GeoLink benchmark, there are 19 simple mappings, including 10 class equivalences, 2 class subsumptions, and 7 property equivalences. Table 5 shows the simple mapping comparison between our algorithm and the matchers that participated in the OAEI 2018. We list the numbers of correctly identified mappings for each matcher and calculate the precision, recall, and f-measure. The confidence value for picking association rules is set to 0.3, since we find it generates the best performance for simple alignments.

Based on the results, our algorithm outperforms other systems on finding the simple mappings in this benchmark. We can argue that leveraging the instance data is a contributing factor, since our algorithm takes advantages of the instance data, while the other alignment systems do not use it. In addition, most traditional alignment systems focus on accurate detection only of 1:1 class equivalences, which limits their performance on this benchmark. The only 1:1 class equivalence that other alignment systems do not find, but our algorithm does, is $\mathsf{gbo{:}Award}(x) \leftrightarrow \mathsf{gmo{:}FundingAward}(x)$. This may also own to the populated instance data. The reason that our algorithm does not achieve 100% precision is that we mistakenly identify that gbo:PortCall is equivalent to gmo:Fix. The correct relationship should be subsumption. This relation can be easily refined by a semi-automated approach in the future.

5.2 Complex Alignment Evaluation

We set the confidence threshold to 1 when running the association rule mining algorithm in order to generate the results described in this section. This is a precision-oriented approach. However, these values can be tuned to fulfill various purposes of alignment systems.

As mentioned previously, in order to assess the quality of a mapping, there are two dimensions that we can look into. First, we can evaluate if the mapping contains the correct entities that should be involved based on the reference alignment. Another dimension is the relationship between the entities, like equivalence and subsumption. Based on this, we break the evaluation procedure down into two subtasks.

Table 6. Similarity for Relationship Identification

Found Relation	Correct Relation	Similarity	Comment
$=$	$=$	1	correct relation
$\subset$	$\subset$	1	correct relation
$\supset$	$\supset$	1	correct relation
$\subset$	$=$	0.8	return less information, but correct
$=$	$\supset$	0.8	return less information, but correct
$\supset$	$=$	0.6	return more information, but incorrect
$=$	$\subset$	0.6	return more information, but incorrect
$\subset$	$\supset$	0.3	incorrect relation
$\supset$	$\subset$	0.3	incorrect relation

(1) Entity Identification: For each entity in the source ontology, the alignment systems will be asked to list all of the entities that are related in some way in the target ontology. For example, referring to the example we used above,

$$\mathsf{Award}(x) \wedge \mathsf{hasSponsor}(x,z) \leftrightarrow \mathsf{FundingAward}(x) \wedge \mathsf{providesAgentRole}(x,y) \\ \wedge \mathsf{SponsorRole}(y) \wedge \mathsf{performedBy}(y,z),$$

the expected output from an alignment system is that `hasSponsor` in the GBO is related to `FundingAward`, `providesAgentRole`, `SponsorRole` and `performedBy` in the GMO and `Award` in the GBO. Based on the two lists of entities from the reference alignment and the matcher, precision, recall, and f-measure can be calculated.

(2) Relationship Identification: In terms of the example above, an alignment system needs to eventually determine that the relationship between the two sides is equivalence. Based on our algorithm, if there is only one association rule holding the information, we consider the relationship to be subsumption. If there are two association rules containing the information for both directions, an equivalence relationship is generated. At this stage, we do not further assess other complex relationships. Table 6 shows the different similarities for different situations. We slightly penalize differently for the situations in finding less information, but all the information returned is correct, and finding more information, but part of the information is incorrect. We do not penalize the incorrect relationship by giving a ZERO value because that would completely neglect the entity identification outputs without considering whether it is a reasonable result or a completely incorrect one. In order to generate the final results, we multiply the results from the entity identification by the penalty of the relations.[10] The formulas for computing the final results are as follows:

```
Relaxed_precision = Precision_entity × Relation_similarity
Relaxed_recall = Recall_entity × Relation_similarity
Relaxed_f-measure = F-measure_entity × Relation_similarity
```

[10] To be accurate, it could also have been better aggregated with other aggregation functions rather than multiplication [7]. But we would not focus on this question in this paper.

Table 7. Comparative Performance of Generating Complex Alignment

Matcher	**1:n Property subsum.**	**m:n Complex equiv.**	**m:n Complex subsum.**
Reference Alignment	5	26	17
Our Algorithm	3	15	7
Relaxed_Precision	0.60	0.90	0.53
Relaxed_Recall	0.36	0.36	0.16
Relaxed_F-measure	0.45	0.51	0.24

Table 7 shows the results of our algorithm. In total there are 48 complex mappings in the reference alignment. For 1:n property subsumption, our algorithm finds 3 mappings that fall into this category. For example, we find that the domain of gbo:hasSampleType is equivalent to gmo:PhysicalSample. However, the correct relationship should be subsumption. So, the final result should be penalized based on Table 6. For m:n complex equivalence, since our default confidence value for complex alignment is 1, the alignment that we found may miss some entities that should exist in the alignment. For example, referring to the example we use in the entity identification, the expected output from the alignment system is that the property hasSponsor in the GBO is related to FundingAward, providesAgentRole, SponsorRole, performedBy in the GMO and Award in the GBO. However, our algorithm misses one entity which is performedBy in the GMO. Errors such as this may of course be easily corrected by human interaction. For m:n complex subsumption, our algorithm does not generate the correct relationships for all the mappings we found. However, overall, our association rule-based algorithm can effectively come up with rather high quality simple and complex alignment automatically.[11]

6 Conclusion

Complex ontology alignment has been discussed for a long time, but relatively little work has been done to advance the state of the art in this field. In this paper, we proposed a complex ontology alignment algorithm based on association rule mining. Our algorithm takes advantage of instance data to mine frequent patterns, which show us which entities in one ontology are related to which entities in the other. Then we apply common simple and complex patterns to arrange these related entities into the formal alignment. We evaluated our system on the complex alignment benchmark from the OAEI and analyzed the results in detail to provide a better understanding of the challenges related to complex ontology alignment research.

There are still some limitations of our algorithm. First, our system relies on instance data for mining the association rules, which is not available for all ontology pairs. However, this could possibly be resolved with automated

[11] All the data and alignment that we use and generate can be accessed via the link http://tiny.cc/rojy4y. We utilize the Apache Spark frequent pattern mining library to generate association rules.

instance data generation to populate common instances into the ontologies or instance matching techniques. Second, we incorporate some common patterns that have been widely accepted in the ontology alignment community in this paper. This could be another limitation, since the set of mapping patterns in our system is likely not comprehensive. However, our algorithm is extensible, more patterns can be easily added in the future as the need arises. Third, it is possible that there are situations that the association rule would fail in term of finding simple property alignment. For example, if there are two properties $livesIn$ and $bornIn$ in source and target ontologies respectively, and the association rules would say if livesIn|Place, then bornIn|Place if they occur frequently. $livesIn$ and $bornIn$ would be considered as equivalent. In this case, there are many different methods that could be applied to improve the performance, like using lexical-based comparison or utilizing external knowledge base to annotate these entities. Fourth, we are collaborating with other benchmark and system developers to enable the comparison and prepare our alignment system to participate in the complex alignment track of the OAEI.

Acknowledgments This material is based upon work supported by the National Science Foundation under Grant No. 1936677.[12]

References

1. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules in large databases. In: VLDB'94, Proc. of 20th International Conference on Very Large Data Bases, September 12-15, 1994, Santiago de Chile, Chile. pp. 487–499 (1994)
2. Algergawy, A., et al.: Results of the ontology alignment evaluation initiative 2018. In: Proceedings of the 13th International Workshop on Ontology Matching, OM@ISWC 2018, Monterey, CA, USA, October 8, 2018. pp. 76–116 (2018)
3. Cheatham, M., et al.: The GeoLink knowledge graph. Big Earth Data **2**(2), 131–143 (2018)
4. David, J., et al.: Association rule ontology matching approach. Int. J. Semantic Web Inf. Syst. **3**(2), 27–49 (2007)
5. Djeddi, W.E., et al.: XMap: results for OAEI 2018. In: Proceedings of the 13th International Workshop on Ontology Matching, OM@ISWC 2018, Monterey, CA, USA, October 8, 2018. pp. 210–215 (2018)
6. Doan, A., et al.: Ontology matching: A machine learning approach. In: Handbook on Ontologies, pp. 385–404 (2004)
7. Ehrig, M., Euzenat, J.: Relaxed precision and recall for ontology matching. In: Integrating Ontologies '05, Proceedings of the K-CAP 2005 Workshop on Integrating Ontologies, Banff, Canada, October 2, 2005 (2005)
8. Euzenat, J.: Semantic precision and recall for ontology alignment evaluation. In: IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence, Hyderabad, India, January 6-12, 2007. pp. 348–353 (2007)
9. Faria, D., et al.: Results of AML participation in OAEI 2018. In: Proceedings of the 13th International Workshop on Ontology Matching, OM@ISWC 2018, Monterey, CA, USA, October 8, 2018. pp. 125–131 (2018)

[12] https://daselab.cs.ksu.edu/projects/spex

10. Galárraga, L.A., et al.: AMIE: association rule mining under incomplete evidence in ontological knowledge bases. In: 22nd International World Wide Web Conference, WWW '13, Rio de Janeiro, Brazil, May 13-17, 2013. pp. 413–422 (2013)
11. Han, J., et al.: Mining frequent patterns without candidate generation: A frequent-pattern tree approach. Data Min. Knowl. Discov. **8**(1), 53–87 (2004)
12. Hertling, S., Paulheim, H.: DOME results for OAEI 2018. In: Proceedings of the 13th International Workshop on Ontology Matching, OM@ISWC 2018, Monterey, CA, USA, October 8, 2018. pp. 144–151 (2018)
13. Jiang, S., et al.: Ontology matching with knowledge rules. T. Large-Scale Data- and Knowledge-Centered Systems **28**, 75–95 (2016)
14. Jiménez-Ruiz, E., Grau, B.C., Cross, V.: LogMap family participation in the OAEI 2018. In: Proceedings of the 13th International Workshop on Ontology Matching, OM@ISWC 2018, Monterey, CA, USA, October 8, 2018. pp. 187–191 (2018)
15. Joshi, A.K., et al.: Logical linked data compression. In: The Semantic Web: Semantics and Big Data, 10th International Conference, ESWC 2013, Montpellier, France, May 26-30, 2013. Proceedings. pp. 170–184 (2013)
16. Laadhar, A., et al.: OAEI 2018 results of POMap++. In: Proceedings of the 13th International Workshop on Ontology Matching, OM@ISWC 2018, Monterey, CA, USA, October 8, 2018. pp. 192–196 (2018)
17. Piatetsky-Shapiro, G.: Discovery, analysis, and presentation of strong rules. In: Knowledge Discovery in Databases, pp. 229–248. AAAI/MIT Press (1991)
18. Portisch, J., Paulheim, H.: ALOD2Vec matcher. In: Proceedings of the 13th International Workshop on Ontology Matching, OM@ISWC 2018, Monterey, CA, USA, October 8, 2018. pp. 132–137 (2018)
19. Ritze, D., et al.: A pattern-based ontology matching approach for detecting complex correspondences. In: Proceedings of the 4th International Workshop on Ontology Matching (OM-2009), Chantilly, USA, October 25, 2009 (2009)
20. Ritze, D., et al.: Linguistic analysis for complex ontology matching. In: Proceedings of the 5th International Workshop on Ontology Matching (OM-2010), Shanghai, China, November 7, 2010 (2010)
21. Shvaiko, P., Euzenat, J.: Ontology matching: State of the art and future challenges. IEEE Trans. Knowl. Data Eng. **25**(1), 158–176 (2013)
22. Stumme, G., Maedche, A.: FCA-MERGE: bottom-up merging of ontologies. In: Proc. of the Seventeenth International Joint Conference on Artificial Intelligence, IJCAI 2001, Seattle, Washington, USA, August 4-10, 2001. pp. 225–234 (2001)
23. Thiéblin, É., et al.: CANARD complex matching system: results of the 2018 OAEI evaluation campaign. In: Proc. of the 13th Int. Workshop on Ontology Matching, OM@ISWC 2018, Monterey, CA, USA, Oct. 8, 2018. pp. 138–143 (2018)
24. Thiéblin, É., et al.: Complex matching based on competency questions for alignment: a first sketch. In: Proc. of the 13th International Workshop on Ontology Matching, OM@ISWC 2018, Monterey, CA, USA, Oct. 8, 2018. pp. 66–70 (2018)
25. Thiéblin, É., et al.: The first version of the OAEI complex alignment benchmark. In: Proc. of the ISWC 2018 Posters & Demonstrations, Industry and Blue Sky Ideas Tracks at (ISWC 2018), Monterey, USA, October 8th - to - 12th, 2018. (2018)
26. Thiéblin, É., et al.: Survey on complex ontology alignment. Semantic Web Journal (2019), to appear
27. Zhou, L., et al.: A complex alignment benchmark: Geolink dataset. In: The Semantic Web - ISWC 2018 - 17th International Semantic Web Conference, Monterey, CA, USA, October 8-12, 2018, Proceedings, Part II. pp. 273–288 (2018)