AROA Results for 2019 OAEI*

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. This paper introduces the results of alignment system AROA in the OAEI 2019 campaign. AROA stands for Association Rule-based Ontology Alignment system. This ontology alignment system can produce simple and complex alignment between ontologies that share common instance data. This is the first participation of AROA in the OAEI campaign, and it produces best performance on one of complex benchmarks (GeoLink).

1 Presentation of the system

1.1 State, purpose, general statement

AROA (Association Rule-based Ontology Alignment) system is aimed to automatically generate simple and complex alignment between two and more ontologies. These ontologies would be required to share common instance data because AROA relies on association rule mining and would require these instances as inputs. After generating a set of association rules, AROA utilizes some simple and complex correspondences that have been widely accepted in Ontology Matching community [4, 6] to further narrow the large number of rules down to more meaningful ones and finally establishes the alignments.

1.2 Specific techniques used

Figure 1 illustrates the overview of AROA alignment system. In this section, we introduce each step of AROA alignment system along with some concepts that we frequently use in AROA system, such as association rule mining, FP-growth algorithm, and complex alignment generation.

Clean Triple. First, AROA extracts all triples as the format of ⟨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 clean the data based on the following criteria: we only keep the triples that contain at least one entity under the source or the target ontology namespace or the triples contain rdf:type information, since our algorithm relies on this information.

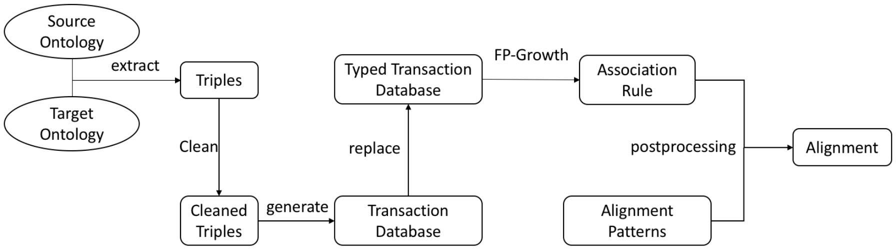

Fig. 1. Overview of AROA Alignment System

Generate Transaction Database. After filtering process, we generate the transaction database as the input for the FP-growth algorithm. Let $I = \{i_1, i_2, \ldots, i_n\}$ be a set of distinct attributes called items. Let $D = \{t_1, t_2, \ldots, t_m\}$ 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. Instance data 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 an instance triple.

Generate Typed Transaction Database. Then we replace the object in the triples with its rdf:type[3] because we focus on generating schema-level (rather than instance-level) mapping rules between two ontologies, and the type

[3] 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 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. 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 3. 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

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 2 and 3 show some examples of the conversion.

Generate Association Rules. 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 [3]. Many algorithms for generating association rules have been proposed, like Apriori [1] and FP-growth algorithm [2]. 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 [2]. 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 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. 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

Table 4. Examples of Association Rules

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

Table 5. The Alignment Pattern Types Covered in AROA System

Pattern	Category
Class Equivalence	1:1
Class Subsumption	1:1
Property Equivalence	1:1
Property Subsumption	1:1
Class by Attribute Type	1:n
Class by Attribute Value	1:n
Property Typecasting Equivalence	1:n
Property Typecasting Subsumption	1:n
Typed Property Chain Equivalence	m:n
Typed Property Chain Subsumption	m:n

likely to be as well. Table 4 shows some examples of association rules generated from the transaction database in Table 1.

Generate Alignment. AROA utilizes some simple and complex correspondences that have been widely accepted in Ontology Matching community to further filter rules [4, 6] and finally generate the alignments. There are totally 10 different types of correspondences that AROA covers in this year. Table 5 lists all the simple and complex alignment correspondences and corresponding category. Since the association rule mining might generate a large number of rules, in order to narrow the association rules down to a smaller set, AROA follows these patterns to generate corresponding alignments. For example, Class by Attribute Type (CAT) is a classic complex alignment pattern. This type of pattern was first introduced in [4]. It 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. Class C_1 is from ontology O_1, and object property op_1 and its range t_1 are from ontology O_2.

Association Rule format: $\mathsf{rdf{:}type}|C_1 \rightarrow \mathsf{op_1}|\mathsf{t_1}$

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 its range gmo:Place. The equivalence relationship can be generated by combining another association rule holding the reverse information. Other simple and complex alignments are also generated by following the same steps.

1.3 Adaptations made for the evaluation

AROA is an instance-based ontology alignment system. Therefore, AROA embeds Apache Jena Fuseki server in the system. The ontologies are first downloaded from the SEALS repository. And then, AROA uploads and stores the ontologies in the embedded Fuseki server, which might take some time for this

Table 6. The Number of Alignments Found on GeoLink Benchmark

Alignment Patterns	**Category**	**Reference Alignment**	**AROA**	
-	-	-	# of Correct Entities	# of Correct Relation
Class Equiv.	1:1	10	10	10
Class Subsum.	1:1	2	1	0
Property Equiv.	1:1	7	5	5
Property Typecasting Subsum.	1:n	5	3	0
Property Chain Equiv.	m:n	26	15	13
Property Chain Subsum.	m:n	17	7	0

step to load large-size ontology pairs. The generated alignments in EDOAL format are available at this link.[4]

2 Results

Since this is the first-year participation, AROA alignment system only evaluates its performance on the GeoLink benchmark. We will evaluate on other benchmarks in the near future. In the GeoLink benchmark, there are 19 simple mappings, including 10 class equivalences, 2 class subsumption, and 7 property equivalences. And there are 48 complex mappings, including 5 property subsumption, 26 property chain equivalences, and 17 property chain subsumption. Table 6 shows alignment patterns and categories in the GeoLink Benchmark and the results of AROA system. We list the numbers of identified mappings for each pattern. There are two dimensions that we can look into the performance. One is the entity identification, which means, given an entity in the source ontology, the system should be able to generate related entities in the target ontology. Another dimension is relationship identification, which the system should detect the correct the relationship between these entities, such as equivalence and subsumption. Therefore, we list the number of correct entities and the number of correct relationships in order to help the reader to understand the strengths and weaknesses of the system. For example, In the Table 6, AROA correctly identifies all 1:1 class equivalence including entity and relationship. However, AROA also finds one class subsumption alignment, which is the class *PortCall* in the GeoLink Base Ontology (GBO) is related to the class *Fix* in the GeoLink Modular Ontology (GMO). However, it outputs the relationship between *PortCall* and *Fix* as equivalence, which it should be subsumption. Therefore, we count the number of correct entities as 1 and number of correct relations as 0. This criterion is also applied to other patterns. In addition, we compare the performance of AROA against other alignment systems in Table 7. And AROA achieved the best performance in terms of relaxed recall and f-measure.[5]

[4] http://oaei.ontologymatching.org/2019/results/complex/geolink/geolink_results.zip

[5] http://oaei.ontologymatching.org/2019/results/complex/geolink/index.html

Table 7. The Performance Comparison on GeoLink Benchmark

Matcher	**AMLC**	**AROA**	**CANARD**	**LogMap**	**LogMapKG**	**LogMapLt**	**POMAP++**
Relaxed_Precision	0.50	0.86	0.89	0.85	0.85	0.69	**0.90**
Relaxed_Recall	0.23	**0.46**	0.39	0.18	0.18	0.25	0.16
Relaxed_F-measure	0.32	**0.60**	0.54	0.29	0.29	0.36	0.26

3 General comments

From the performance comparison, only AROA and CANARD [5] can generate almost correct complex alignment, which means some alignments found by these two systems may not be completely correct, but it can be easily improved by semi-automated fashion. For example, the system can produce correct entities that should be involved in a complex alignment, but it doesn't output the correct relationship. Another situation is that the system can detect the correct relationship but fails to find all the entities. Based on these situations, we will investigate the incorrect alignments and improve the algorithm to find the relationship and entities as accurate as possible.

4 Conclusions

This paper introduces the AROA ontology alignment system and its preliminary results in the OAEI 2019 campaign. This year, AROA evaluates its performance on GeoLink benchmark and achieves the best performance in terms of recall and f-measure. We will continue to evaluate AROA on other benchmarks and improve the algorithm in the near future.

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), http://www.vldb.org/conf/1994/P487.PDF
2. Han, J., et al.: Mining frequent patterns without candidate generation: A frequent-pattern tree approach. Data Min. Knowl. Discov. **8**(1), 53–87 (2004). https://doi.org/10.1023/B:DAMI.0000005258.31418.83, https://doi.org/10.1023/B:DAMI.0000005258.31418.83
3. Piatetsky-Shapiro, G.: Discovery, analysis, and presentation of strong rules. In: Knowledge Discovery in Databases, pp. 229–248. AAAI/MIT Press (1991)
4. Ritze, D., Meilicke, C., Sváb-Zamazal, O., Stuckenschmidt, H.: A pattern-based ontology matching approach for detecting complex correspondences. In: Proceedings of the 4th International Workshop on Ontology Matching (OM-2009) collocated with the 8th International Semantic Web Conference (ISWC-2009) Chantilly, USA, October 25, 2009 (2009), http://ceur-ws.org/Vol-551/om2009_Tpaper3.pdf
5. 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), http://ceur-ws.org/Vol-2288/oaei18_paper4.pdf

6. 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). https://doi.org/10.1007/978-3-030-00668-6_17, https://doi.org/10.1007/978-3-030-00668-6_17