Debugging Multi-Property Correspondences in Ontology Alignment Scenarios

Alexandre Gouveia[1], João Rocha[1], Paulo Martins[2], and Nuno Silva[1]

GECAD – Knowledge Engineering and Decision Support Research Group
[1]Instituto Superior de Engenharia do Porto, Porto, Portugal
{aas,jsr,nps}@isep.ipp.pt
[2]Universidade de Trás-os-Montes e Alto Douro, Vila Real, Portugal
pmartins@utad.pt

Abstract: Data transformation between ontology repositories demands great quality alignments and as such ambiguous correspondences must be identified and corrected beforehand. In previous work we characterized 1:1 ambiguous ontology matching scenarios through ten dimensions and defined a set of resolution strategies. The main novelty of this paper is the systematization of complex scenarios (1:n, n:1 or n:m) and their transformation to 1:1 scenarios, by aggregating the n properties of each candidate correspondence into a virtual single entity. This analytical approach allows profiting from the previously defined resolution strategies.

Keywords: Ontology: Alignment: Correspondences

1 Motivation

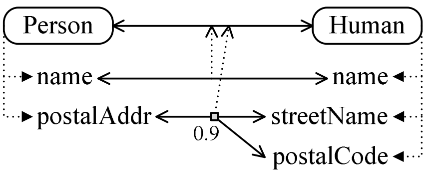
Automatic alignment systems aim to create meaningful and correct ontology alignments (i.e. a set of correspondences between entities of two or more ontologies) [1]. However, the results obtained are below the requirement for ontology mediation (e.g. data integration, migration, data transformation) [1–3], especially because there is not a direct and unique relation between these automatic alignments and the alignments that allow data integration. On the other hand, manual systems, e.g. MAFRA Toolkit, MapForce, Neon Toolkit and Snoogle use complex, time-consuming and yet error prone mapping processes that require extensive and profound (human) knowledge of the domain. It is therefore necessary to bridge the gap between automatically generated and data-integration-ready alignments.

The correspondences automatically-generated by the matchers are of two types: (i) concept-correspondences in the form of $(\{c1\},\{c2\},r,n)$, where $c1$ and $c2$ are concepts; and (ii) property-correspondences in the form of $(\{p1\},\{p2\},r,n)$, where $p1$ and $p2$ are properties. In both cases r is the relation held between the entities (e.g. equivalence, narrow, concatenation) and n is the confidence value in the relation.

Notice that although a property can have multiple domain concepts they are not specified in the property-correspondence. What is more, there is no guarantee that their domain concepts are mapped in a concept-correspondence, despite this is mandatory [4]. Such automatically-generated alignments allow multiple interpretations giving rise to ambiguities during the transformation process (e.g. what domain concept of the property should be considered). Formally, the debugging process must provide a data-integration-ready alignment, characterized by:

- each property-correspondence defines the domain concepts of the properties;
- a concept-correspondence exists for every pair of domain concepts defined in a property-correspondence.

Consider the alignment of Fig. 1. The correspondence between Person and Human is responsible for transforming every instance of Person into Human, and the correspondence between O1:name and O2:name transforms (copy) the value of name of every Person into the name of the respective Human. The value of postalAddr is split into streetName and postalCode.



$$(\{Person\}, \{Human\}, =, 1)$$
$$(\{Person.name\}, \{Human.name\}, =, 1)$$
$$\left(\begin{array}{c}\{Person.postalAddr\},\\ \{Human.streetName, Human.postalCode\},\\ split, 0.9\end{array}\right)$$

Fig. 1. Alignment between two ontologies.

There are two types of property-correspondences: single-domain and multi-domain. In single-domain correspondences all the properties of the correspondence have a common domain concept. In a multi-domain correspondence at least two properties of the correspondence have different domain concepts. An example of a multi-domain property-correspondence is $(\{c1.p1, c2.p2\}, \{cA.pA\}, r, n)$ where $c1$ is the domain concept of property $p1$ and $c2$ is the domain concept of $p2$. The remainder of the paper however addresses single-domain property-correspondences only.

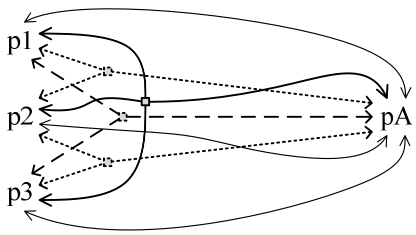
Transversely to all of the analyzed state-of-the-art approaches [5], is the observation that they are not focused on correctly improving the coverage but instead on solving the problems. In fact none of the related work considers the possibility of adding new correspondences between concepts to solve an ambiguity found in a property-correspondence.

2 Complex Scenarios

The problems identified in previous work [4], [5] address 1:1 property-correspondences. However, most of the automatically-generated correspondences correspond to more complex correspondences, including multiplicity of the cardinality and relations other than “equivalence” (cf. Fig. 1). A set of two or more correspondences between properties where one of the (source or target) properties is mapped to two or more (source or target) properties is called a Complex scenario. In terms of cardinality there are three types of Complex scenarios: 1:n, n:1 and n:m. When cardinality is n:m, the scenario is a combination of both 1:m and n:1 scenarios.

The semantics of each 1:n (or n:1) matching scenario is not unique and thus several correspondences are possible. Different semantics correspond to different candidate correspondences. Table 1 illustrates the semantics/correspondences of a 3:1 scenario.

Table 1. Semantics of a 3:1 complex scenario.

Complex scenario	Candidate correspondences obtained from Complex scenario
$(\{p1\},\{pA\})$ $(\{p2\},\{pA\})$ $(\{p3\},\{pA\})$	$(\{p1\},\{pA\})$ $(\{p2\},\{pA\})$ $(\{p3\},\{pA\})$ $(\{p1,p2\},\{pA\})$ $(\{p1,p3\},\{pA\})$ $(\{p2,p3\},\{pA\})$ $(\{p1,p2,p3\},\{pA\})$ 

Analyzing and generalizing the examples to 1:n (or n:1) scenarios, the number of candidate correspondences is given by $\sum_{i=1}^{n} C_{i}^{n}$, such that $C_{i}^{n} = \frac{n!}{i!(n-i)!}$. This corresponds to (i) n 1:1; and (ii) $\sum_{i=2}^{n} C_{i}^{n}$ 1:m (or m:1) candidate correspondences.

Some of the candidate correspondences are (or might be) impossible and some are more admissible than others depending on their domain concepts and existing correspondences between their domain concepts. Such analysis will allow the reduction of possible semantic interpretations and therefore disambiguate the matching scenarios.

3 Transforming 1:n and n:1 Correspondences to 1:1

We suggest addressing each 1:n (or n:1) candidate correspondence as 1:1 correspondence, by aggregating the n properties into a virtual single entity. This would allow the processing of each candidate correspondence according to the systematization addressed in previous work [4], [5].

Fig. 2 illustrates the valid single-domain candidate correspondences of a n:1 Complex scenario. Observe that besides the illustrated valid single-domain candidate correspondences, other candidates (e.g. $(\{p1,p3\},\{pA\})$) were discarded because their properties do not have a common domain concept. For example, the $(\{p1,p2\},\{pA\})$ candidate correspondence has six valid single-domain candidate correspondences:

$$(\{c1.p1, c1.p2\},\{cA.pA\}),\ (\{c1.p1, c1.p2\},\{cB.pA\}),\ (\{c1.p1, c1.p2\},\{cC.pA\}),$$
$$(\{c2.p1, c2.p2\},\{cA.pA\}),\ (\{c2.p1, c2.p2\},\{cB.pA\}),\ (\{c2.p1, c2.p2\},\{cC.pA\})$$

Besides identifying the properties' domain concept, it is necessary that a correspondence exists between the domain concepts. Fig. 3 illustrates a possible resolution for the candidate correspondence $(\{p1,p2\},\{pA\})$, creating correspondences between the direct-domain concepts and specifying relationships between the property-correspondence and the created concept-correspondences.

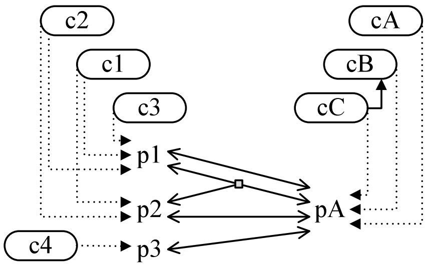


Fig. 2. Valid single-domain candidates.

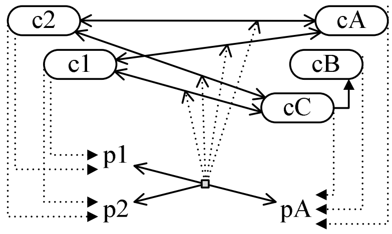


Fig. 3. Possible resolution of the $(\{p1, p2\}, \{pA\})$ scenario of Fig. 2.

4 Summary and Future Work

This paper suggests addressing 1:n (or n:1) scenarios as 1:1, by aggregating the n properties of each candidate correspondence into a virtual single entity, thus promoting the application of the previously devised disambiguation strategies.

Yet, this approach conducts to the blind adoption of corrective actions as they do not consider the semantics of each concrete scenario in hand. In consequence, we will focus our efforts in profiting from an iterative alignment debugging process that feeds and re-executes the matching algorithms with the generated ambiguous scenarios and resolutions strategies, for proposing solutions to the identified problems.

Acknowledgements

This work is supported by FEDER Funds through the “Programa Operacional Factores de Competitividade - COMPETE” program and by National Funds through FCT “Fundação para a Ciência e a Tecnologia” under the project: FCOMP-01-0124-FEDER-PEst-OE/EEI/UI0760/201 and through the project World Search (QREN11495) of FEDER.

References

1. Euzenat, J., Shvaiko, P.: Ontology matching, Heidelberg, Germany: Springer-Verlag (2007)
2. de Bruijn, J., Ehrig, M., Feier, C., Martíns-Recuerda, F., Scharffe, F., Weiten, M.: Ontology mediation, merging, and aligning. In: Semantic Web Technologies (2006).
3. Halevy, A., Rajaraman, A., Ordille, J.: Data integration: the teenage years. In: Very Large Data Bases, Seoul, Korea, pp. 9–16 (2006).
4. Gouveia, A., Silva, N., Rocha, J.: Automatic ontology alignment disambiguation based on ontological structural dimension. In: Proceedings of the 14th International Conference on Enterprise Information Systems, Wroclaw, Poland (2012).
5. Gouveia, A., Silva, N., Rocha, J., Martins, P.: Iterative ontology alignment debugging using a scenario- and strategy-driven approach. In: Proceedings of the 11th IEEE International Workshop on Web Semantics and Information Processing, Vienna, Austria (2012).