Complex Matching of RDF Datatype Properties

Bernardo P. Nunes[1], Alexander Mera[1], Marco A. Casanova[1], Karin K. Breitman[1]
Luiz André P. Paes Leme[2]

[1]Department of Informatics – PUC-Rio – Rio de Janeiro, RJ – Brazil
{bnunes, acaraballo, casanova, karin}@inf.puc-rio.br
[2]Instituto de Computação – UFF – Rio de Janeiro, RJ – Brazil
lapaesleme @ ic.uff.br

Abstract. Property mapping is a fundamental component of ontology matching, and yet there is little support that goes beyond the identification of single property matches. Real data often requires some degree of composition, trivially exemplified by the mapping of *FirstName, LastName* to *FullName* on one end, to complex machings, such as parsing and pairing symbol/digit strings to *SSN* numbers, at the other end of the spectrum. In this paper, we briefly introduce a two-phase instance-based technique for complex datatype property matching.

Keywords: Ontology Matching, Genetic Programming, Mutual Information.

1 Introduction

Ontology matching is a fundamental problem in many applications areas [1]. Using OWL concepts, by *datatype property matching* we mean the special case of matching datatype properties from two classes.

Very briefly, an *instance* of a datatype property p is a triple of the form (s,p,l), where s is a resource identifier and l is a literal. A *datatype property matching* from a *source* class S to a *target* class T is a partial relation μ between sets of datatype properties of S and sets of datatype properties of T. We say that a match $(\boldsymbol{A},\boldsymbol{B}) \in \mu$ is $m{:}n$ iff $\boldsymbol{A}$ and $\boldsymbol{B}$ contain m and n properties, respectively. A match $(\boldsymbol{A},\boldsymbol{B}) \in \mu$ should be accompanied by one or more *datatype property mappings* that indicate how to construct instances of the properties in $\boldsymbol{B}$ from instances of the properties in $\boldsymbol{A}$. A match $(\boldsymbol{A},\boldsymbol{B}) \in \mu$ is *simple* iff it is *1:1* and the mapping is a simple translation; otherwise, it is *complex*.

In this paper, we briefly introduce a two-phase, instance-based datatype property matching technique that is able to find complex *n:1* datatype property matches and to construct the corresponding property mappings. The technique extends the ontology matching process described in [2] to include complex matches between sets of datatype properties and is classified as instance-based since it depends on sets of instances.

2 The Two-Phase Property Matching Technique

Given two sets, ***s*** and ***t***, that contain instances of the datatype properties of the source class *S* and the target class *T*, respectively, the first phase of the technique constructs the Estimated Mutual Information matrix [2,3] of the datatype property instances in ***s*** and the datatype property instances in ***t***, which intuitively measures the amount of related information of the observed property instances. This phase possibly identifies simple datatype property matches. For example, it may detect that the *eMail* datatype property of one class matches the *ElectronicAddress* datatype property of the other class. The first phase may also suggest, for the second phase, sets of datatype properties that may match in more complex ways, thereby reducing the search space.

The second phase uses a genetic programming approach [4] to find complex *n:1* datatype property matches. For example, it may discover that the *FirstName* and *LastName* datatype properties of the source class matches the *FullName* datatype property of the target class, and return a property mapping function that concatenates the values of *FirstName* and *LastName* (of the same class instance) to generate the *FullName* value. The reason for adopting genetic programming is two-fold: it reduces the cost of traversing the search space; and it may be used to generate complex mappings between datatype property sets.

3 Conclusion

In this paper, we briefly described an instance-based, property matching technique that follows a two-phase strategy. The first phase constructs the Estimated Mutual Information matrix of the property values to identify simple property matches and to suggest complex matches, while the second phase uses a genetic programming approach to detect complex property matches and to generate their property mappings. Our early experiments suggest that the technique is a promising approach to construct complex property matches, a problem rarely addressed in the literature. Full details can be found in [5].

Acknowledgements. This work was partly supported by CNPq, under grants 473110/2008-3 and 557128/2009-9, by FAPERJ under grant E-26/170028/2008, and by CAPES under grant CAPES/PROCAD NF 21/2009.

References

1. Euzenat, J., Shvaiko, P. *Ontology matching*. Springer-Verlag (2007).
2. Leme, L. A. P. P., Casanova, M. A., Breitman, K. K., Furtado, A. L. Instance-Based OWL Schema Matching, Lectures Notes in Business Info. Proc., vol. 24, 2009, pp.14-25.
3. Leme, L. A. P. P., Brauner, D. F., Breitman, K. K., Casanova, M. A., Gazola, A. Matching Object Catalogues, Innov. in Sys. and Soft. Eng. Springer, 4(4), 2008, pp. 315-328.
4. Koza, J. *Genetic Programming*. The MIT press, 1998.
5. Nunes, B. P., Mera, A., Casanova, M. A., Breitman, K. K., Leme, L. A. P. P. Complex Matching of RDF Datatype Properties. MCC12/11, Dept Informatics, PUC-Rio (September 2011).