Instance-Based Property Matching in Linked Open Data Environment

Cheng Xie[1], Dominique Ritze[2], Blerina Spahiu[3], and Hongming Cai[1]

[1] Shanghai Jiao Tong University
[2] University of Mannheim
[3] University of Milano-Bicocca
chengxie@sjtu.edu.cn

Abstract. Instance matching frameworks that identify links between instances, expressed as `owl:sameAs` assertions, have achieved a high performance while the performance of property matching lags behind. In this paper, we leverage `owl:sameAs` links and show how these links can help for property matching.

Keywords: Property matching, Instance-based matching, Linked Open Data

Introduction The performance of ontology matching systems on property matching lags significantly behind that on class and instance matching [1]. Current state-of-the-art techniques achieve a high performance on instance matching which focus on finding `owl:sameAs` links between LOD datasets [2]. These linked instances give an important information to the property matching process which we further explore in this paper. We argue that `owl:sameAs` instance pairs share similar values on similar properties. For this issue, we investigate to which extent we can automatically find matching properties by exploiting `owl:sameAs` instance pairs.

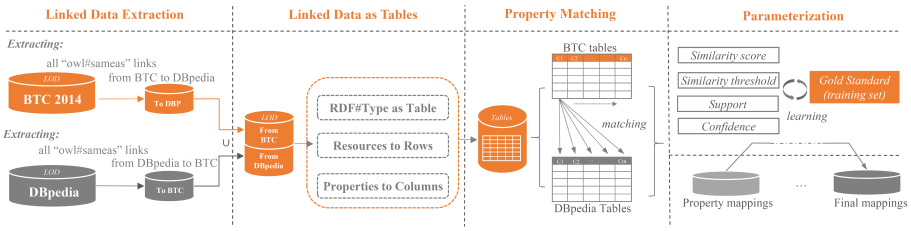

Fig. 1. General matching pipline

Approach Figure1 shows a concrete example of matching DBpedia to BTC2014[4]. The proposed approach has four steps which are described in detail below.

Linked Data Extraction: As first step, we extract all `owl:sameAs` triples whose subject is an instance in BTC2014 while the object is an instance in DBpedia. With the same heuristic we extract `owl:sameAs` links from DBpedia to BTC2014 so we have a complete set of linked instances between these datasets.

[4] http://km.aifb.kit.edu/projects/btc-2014/

Linked Data as Tables: The table generation approach is based on DBpediaAsTable[5] with proper modifications to fit BTC2014 triples. We create one table for each `rdf:type` and place instances of that type in rows, while the columns contain information about their properties. In practice, large tables are separated into several small tables by the limitation of 500 rows while columns are filtered by the density limitation which should be greater than 20%.

Property Matching: We argue that "`owl:sameAs` instances share similar values on similar properties". Once we obtain the `owl:sameAs` instances and similar values, similar properties could be inferred. Similar values are detected by computing similarity measures on literal, numeric and date cells. Afterwards, we can infer similar properties.

Parametrization: The final property correspondences are selected from a candidate set that is obtained from the property matching in last step. The selection is made by filtering property pairs using support threshold *su* and confidence threshold *co*. Property pair (p_1,p_2) holds with support *su* if *su*% of the `owl:sameAs` instances involved with p_1 or p_2 contain both p_1 and p_2. Property pair (p_1,p_2) holds with confidence *co* if *co*% of value pairs on (p_1,p_2) share similar values. We divide our gold standard into a learning set and a testing set. A genetic learning algorithm is applied on the learning set to obtain the proper values for *su* and *co*.

Result. We use three string-based metrics, Jaccard, Levenshtein and ExactEqual as baselines to compare with our approach. All metrics are applied on the testing set to find equivalent properties between BTC2014 and DBpedia. The results and the comparison is shown in Table 1.

Experiments	True Positive	False Positive	GS	Pre	Rec	F1
Instance-based property matching	84	23	85	0.785	0.988	0.875
Levenshtein	52	52	85	0.5	0.612	0.550
Jaccard	52	91	85	0.364	0.612	0.456
ExactEqual	32	0	85	1.0	0.376	0.547

Table 1. The results on property matching between BTC2014 and DBpedia.

The proposed approach can effectively match the property pairs which share similar values such as "landArea" with "areaTotal" and "diedIn" with "deathPlace". However, similar values also lead to wrong matchings such as "happenedOnDate" with "date", "capital" with "largestCity" and "hasPhotoCollection" with "label" which require more semantic matching on property labels than on their values.

References

[1] M. Cheatham and P. Hitzler. The properties of property alignment. In *Proc. of the 9th Int.'l Workshop on Ontology Matching (OM)*, pages 13–24, 2014.

[2] M. Nentwig, M. Hartung, A.-C. N. Ngomo, and E. Rahm. A Survey of Current Link Discovery Frameworks. *Semantic Web Journal*, 2015.

[5] http://wiki.dbpedia.org/services-resources/downloads/dbpedia-tables