Pushing the Limits of Instance Matching Systems: A Semantics-Aware Benchmark for Linked Data

Tzanina Saveta
ICS - FORTH, Greece
jsaveta@ics.forth.gr

Evangelia Daskalaki
ICS - FORTH, Greece
eva@ics.forth.gr

Giorgos Flouris
ICS - FORTH, Greece
fgeo@ics.forth.gr

Irini Fundulaki
ICS - FORTH, Greece
fundul@ics.forth.gr

Melanie Herschel
IPVS - University of Stuttgart, Germany
melanie.herschel@ipvs.uni-stuttgart.de

Axel-Cyrille Ngonga Ngomo
IFI/AKSW
University of Leipzig, Germany
ngonga@informatik.uni-leipzig.de

ABSTRACT

The architectural choices behind the Data Web have led to the publication of large interrelated data sets that contain different descriptions for the same real-world objects. Due to the mere size of current online datasets, such duplicate instances are most commonly detected (semi-)automatically using instance matching frameworks. Choosing the right framework for this purpose remains tedious, as current instance matching benchmarks fail to provide end users and developers with the necessary insights pertaining to how current frameworks behave when dealing with real data.

In this poster, we present the Semantic Publishing Instance Matching Benchmark (SPIMBENCH) which allows the benchmarking of instance matching systems against not only structure-based and value-based test cases, but also against semantics-aware test cases based on OWL axioms. SPIMBENCH features a scalable data generator and a weighted gold standard that can be used for debugging instance matching systems and for reporting how well they perform in various matching tasks.

1. INTRODUCTION

Instance matching (IM), refers to the problem of identifying instances that describe the *same real-world object* [1, 2, 3, 4, 5]. For Linked Data, novel IM techniques have recently been proposed [6, 7], surveyed in [8].

With the increasing adoption of Semantic Web Technologies and the publication of large interrelated RDF data sets and ontologies that form the Linked Data Cloud[1], it is crucial to develop IM techniques adapted to this setting that is characterized by an unprecedented number of sources across which to detect matches, a high degree of heterogeneity both at the schema and instance level, and rich semantics that accompany schemas defined in terms of expressive languages such as OWL, OWL 2, and RDFS.

This paper presents the *Semantic Publishing Instance Matching Benchmark*[9], SPIMBENCH for short, a novel IM benchmark for the assessment of IM techniques for RDF data with an associated schema. Essentially, SPIMBENCH proposes and implements: *(i)* a set of *test cases* based on transformations that distinguish different types of matching entities, *(ii)* a scalable data generator, *(iii)* a gold standard documenting the matches that IM systems should find, and *(iv)* evaluation metrics.

SPIMBENCH extends the state-of-the-art IM benchmarks for RDF data in three main aspects: it allows for systematic scalability testing, supports a wider range of test cases implemented by means of *transformations* and provides an enriched gold standard. SPIMBENCH is the first benchmark to support *semantics-aware* test cases that go beyond the standard RDFS constructs. More precisely, it is the first benchmark to support the OWL constructs for *instance (in)equality*, class and property *equivalence* and *disjointness*, *property constraints*, as well as *complex class definitions*. SPIMBENCH also supports *simple* test cases (implemented using the aforementioned transformations applied on different triples pertaining to the same instance), as well as *complex* test cases (implemented by combinations of individual transformations on the same triple).

2. SPIMBENCH

2.1 Transformations

In SPIMBENCH we propose a set of *value-based*, *structure-based*, and *semantics-aware* test cases, as well as *simple* and *complex* ones.

Value-based test cases refer to scenarios implemented using *transformations* on *instance data type properties* and refer to mainly typographical errors and the use of different data formats. Each transformation takes as input a *data type property* as specified in SPIMBENCH's schema and a *severity* that determines how important this modification is. We used SWING [10] to implement the supported transformations, which are a superset of those considered in the

[1] http://linkeddata.org/

WWW 2015 Companion, May 18–22, 2015, Florence, Italy.
ACM 978-1-4503-3473-0/15/05.
http://dx.doi.org/10.1145/2740908.2742729.

state of the art IM benchmarks. *Structure-based* test cases are based on the use of transformations applied on properties of instances such as *splitting*, *aggregation*, *deletion*, and *addition*.

In addition, SPIMBENCH is the first benchmark to support *semantics-aware* test cases that go beyond the standard RDFS constructs. These are primarily used to examine if the matching systems take into consideration OWL and OWL 2 axioms to discover matches between instances that can be found *only* when considering schema information. The axioms that we consider in SPIMBENCH are:

- *instance (in)equality* (`owl:sameAs`, `owl:differentFrom`)
- *class* and *property equivalence* (`owl:equivalentClass`, `owl:equivalentProperty`)
- *class* and *property disjointness* (`owl:disjointWith`, `owl:AllDisjointClasses`, `owl:propertyDisjointWith`, `owl:AllDisjointProperties`)
- *class* and *property* hierarchies (`rdfs:subClassOf`, `rdfs:subPropertyOf`)
- *property constraints* (`owl:FunctionalProperty`, `owl:InverseFunctionalProperty`)
- *complex class definitions* (`owl:unionOf`, `owl:intersectionOf`)

Furthermore, in SPIMBENCH we consider combinations of the aforementioned test cases. We distinguish between *simple* test cases based on value, structural and semantics-aware test cases, applied on different triples pertaining to one instance. We also consider *complex test cases* that are based on combinations of test cases applied to a *single* triple. A sample of generated datasets can be found at [9], and are omitted due to lack of space.

2.2 Data Generation

The SPIMBENCH data extends the one proposed by the Semantic Publishing Benchmark (SPB) [11] and produces RDF descriptions of *creative works* that are valid instances of classes of the ontologies provided by BBC. This class collects all RDF descriptions of creative works (also called *journalistic assets*) created by the publisher's editorial team. In generating test datasets to be used for IM, we first generate a *synthetic source dataset*, ensuring that this dataset does not contain any matches itself. Next, we generate matches and non-matches to entities of the source dataset to cover the test cases described above. As a result, we obtain a *synthetic target dataset* that contains matches that IM methods should identify. Our data generation process allows the generation of arbitrary large datasets, thus supporting the evaluation of both the scalability and the matching quality of an IM system.

2.3 Gold Standard

To improve the debugging of instance matching tools and algorithms, we assign weights to each pair of instances that should be matched. In essence, the weight of a match (u_i, u_i') quantifies how easy it is to detect this match automatically. We adopt an information-theoretical approach to compute the weight of (u_i, u_i') by measuring the information loss that results from applying transformations to the source data to generate the target data. The basic idea behind our approach is to apply multi-relational learning (MRL) $\mathcal{L}$ to the input knowledge base K and the transformed knowledge base K'. By comparing the description of u_i in $\mathcal{L}(K)$ and u_i' in $\mathcal{L}(K')$, we should then be able to quantify how much information was lost through the transformation of K to K'. We implement this insight in the current version of SPIMBENCH by using RESCAL [12] as MRL approach.

3. FUTURE WORK

We ran our benchmark against a well-known IM system (LogMap [13]) which generally performed well, except in cases where multiple semantics-aware transformations exist (see [9] for details); in the future, we plan to run the benchmark in more state-of-the-art IM systems. We are currently working on a domain-independent instance matching test case generator for Linked Data, whose aim is to take any RDF dataset as source and to produce a target dataset that will implement the test cases discussed earlier. We are also studying how we can define more sophisticated metrics that take into account the difficulty (weight) of the correctly identified matches, to be used in tandem with the standard precision and recall metrics.

Acknowledgements. This work was partially supported by the FP7 European Project Linked Data Benchmark Council (LDBC) (#317548).

4. REFERENCES

[1] I. Bhattacharya and L. Getoor. *Entity resolution in graphs. Mining Graph Data*. Wiley and Sons, 2006.

[2] A. K. Elmagarmid, P.G. Ipeirotis, and V.S. Verykios. Duplicate Record Detection: A Survey. *TKDE*, 19(1), 2007.

[3] C. Li, L. Jin, and S. Mehrotra. Supporting efficient record linkage for large data sets using mapping techniques. In *WWW*, 2006.

[4] J. Noessner, M. Niepert, C. Meilicke, and H. Stuckenschmidt. Leveraging Terminological Structure for Object Reconciliation. In *ESWC*, 2010.

[5] P. Christen. *Data Matching - Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection*. Data-Centric Systems and Applications. Springer, 2012.

[6] R. Isele, A. Jentzsch, and C. Bizer. Silk Server - Adding missing Links while consuming Linked Data. In *COLD*, 2010.

[7] A.-C. Ngonga Ngomo and Soren Auer. LIMES - A Time-Efficient Approach for Large-Scale Link Discovery on the Web of Data. *IJCAI*, 2011.

[8] K. Stefanidis, V. Efthymiou, M. Herschel, and V. Christophides. Entity resolution in the web of data. In *WWW, Companion Volume*, 2014.

[9] SPIMBENCH description. Available at http://www.ics.forth.gr/isl/spimbench/.

[10] A. Ferrara, S. Montanelli, J. Noessner, and H. Stuckenschmidt. Benchmarking Matching Applications on the Semantic Web. In *ESWC*, 2011.

[11] I. Fundulaki, N. Martinez, R. Angles, B. Bishop, and V. Kotsev. D2.2.2 Data Generator. Technical report, Linked Data Benchmark Council, 2013. Available at http://ldbc.eu/results/deliverables.

[12] D. Krompass, M. Nickel, X. Jiang, and V. Tresp. Non-Negative Tensor Factorization with RESCAL. In *TML*, 2013.

[13] E. Jiménez-Ruiz and B. C. Grau. Logmap: Logic-based and scalable ontology matching. In *ISWC*, 2011.