LANCE: A Generic Benchmark Generator for Linked Data

Tzanina Saveta[1], Evangelia Daskalaki[1], Giorgos Flouris[1], Irini Fundulaki[1], and Axel-Cyrille Ngonga Ngomo[2]

[1] Institute of Computer Science-FORTH* Greece,
[2] IFI/AKSW, University of Leipzig, Germany

Abstract. Identifying duplicate instances in the Data Web is most commonly performed (semi-)automatically using instance matching frameworks. However, 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 demo paper, we present LANCE, a domain-independent instance matching benchmark generator for Linked Data. LANCE is the first benchmark generator for Linked Data to support *semantics-aware* test cases that take into account complex OWL constructs in addition to the standard test cases related to structure and value transformations. LANCE supports the definition of matching tasks with varying degrees of difficulty and produces a weighted gold standard, which allows a more fine-grained analysis of the performance of instance matching tools. It can accept as input *any* linked dataset and its accompanying schema to produce a target dataset implementing test cases of varying levels of difficulty. In this demo, we will present the benchmark generation process underlying LANCE as well as the user interface designed to support LANCE users.

1 Introduction

Instance matching (IM), refers to the problem of identifying instances that describe the *same real-world object*. With the increasing adoption of Semantic Web technologies and the publication of large interrelated RDF datasets and ontologies that form the Linked Data (LD) Cloud a number of IM techniques adapted to this setting have been proposed [1,2,3]. Clearly, the large variety of IM techniques requires their comparative evaluation to determine which technique is best suited for a given application. Assessing the performance of these systems generally requires *well-defined and widely accepted benchmarks* to determine the weak and strong points of the methods or systems in addition to motivate the development of better systems to overcome the identified weak points. Hence, suited benchmarks help push the limit of existing systems [4,5,6,7,8], advancing both research and technology.

In this paper[1], we describe LANCE, a flexible, generic and domain-independent benchmark generator for IM systems. LANCE supports a large variety of value,

* This work was partially supported by the EU FP7 projects LDBC (FP7-ICT-2011-8 #317548) and H2020 PARTHENOS (#654119).

[1] This demo paper is a companion paper to the accepted ISWC research paper [8].

structure based and semantics-aware transformations with varying degrees of difficulty. The results of these transformations can be recorded in the form of a weighted gold standard that allows a more fine-grained analysis of the performance of instance matching tools. This paper focuses on describing the interface that allows users to generate a benchmark by providing the different parameters that determine the characteristics of the benchmark (source datasets, types and severity of transformations, size of the generated dataset and further configurations such as new namespace for the transformed instances, output date format and other). Details on the different types of transformations, our weighted gold standard and metrics, as well as the evaluation of our system can be found in [8]. Our demo can be found at `http://tinyurl.com/pvex9hu`.

2 LANCE Approach

LANCE [8] is a flexible, generic and domain-independent benchmark generator for IM systems whose main features are:

Transformation-based test cases. LANCE supports a set of *test cases* based on *transformations* that distinguish different types of matching entities. Similarly to existing IM benchmarks, LANCE supports *value-based* (typos, date/number formats, etc.) and *structure-based* (deletion of classes/properties, aggregations, splits, etc.) test cases. LANCE is the *first benchmark generator* to support *semantics-aware* test cases that go beyond the standard RDFS constructs and allow testing the ability of IM systems to use the semantics of RDFS/OWL axioms to identify matches and include tests involving *instance (in)equality*, class and property *equivalence* and *disjointness*, *property constraints*, as well as *complex class definitions*. LANCE also supports *simple combination (SC)* test cases (implemented using the aforementioned transformations applied on different triples pertaining to the same instance), as well as *complex combination (CC)* test cases (implemented by combinations of individual transformations on the same triple).

Similarity score and fine-grained evaluation metrics. LANCE provides an enriched, *weighted gold standard* and related evaluation metrics, which allow a more fine-grained analysis of the performance of systems for tests with varying difficulty. The gold standard indicates the matches between source and target instances. In particular, each match in the gold standard is enriched with annotations specific to the test case that generated each pair, i.e., the type of test case it represents, the property on which a transformation was applied, and a *similarity score* (or *weight*) of the pair of reported matched instances that essentially quantifies the difficulty of finding a particular match. This detailed information allows LANCE to provide more detailed views and novel evaluation metrics to assess the completeness, soundness, and overall matching quality of an IM system on top of the standard precision/recall metrics. Therewith, LANCE provides fine-grained information to support debugging and extending IM systems.

High level of customization and scalability testing. LANCE provides the ability to build benchmarks with different characteristics on top of any input dataset, thereby allowing the implementation of diverse test cases for different

domains, dataset sizes and morphology. This makes LANCE highly customizable and domain independent; it also allows systematic scalability testing of IM systems, a feature which is not available in most state-of-the-art IM benchmarks.

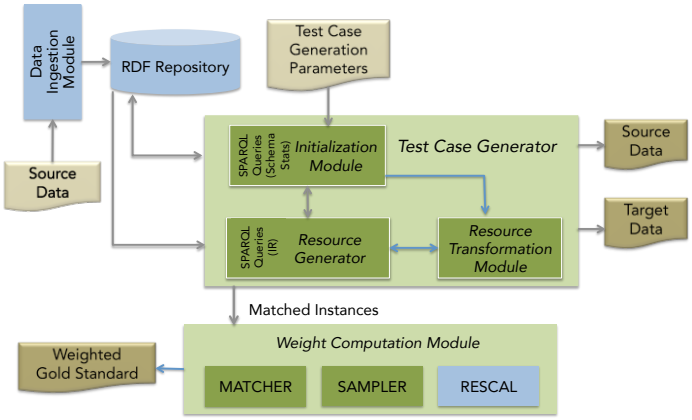

Fig. 1. LANCE System Architecture

3 Implementation and Demonstration

In the following, we present the functionality which we will also explain during the demo. Architecturally, LANCE consists of two components: (i) an *RDF repository* that stores the source datasets, and (ii) a *test case generator*, which takes a source dataset as input and produces a target dataset. The target dataset is generated by using some or all of the various *test cases* implemented by LANCE according to the configuration parameters specified by the user (see Figure 1).

The test case generator consists of the *initialization*, *resource generator* and the *resource transformation* modules. The first reads the generation parameters and retrieves the schema that will be used for producing the target dataset. The *Resource Generator* uses this input to retrieve instances of those schema constructs and to pass them (along with the configuration parameters) to the *resource transformation module*, which creates and stores one transformed instance per source instance. Once LANCE has performed all the requested transformations, the *Weight Computation Module* calculates the similarity scores of the produced matches.

We have developed a Web application on top of LANCE accessible at `http://tinyurl.com/pvex9hu` that allows one to produce benchmarks by selecting the source dataset (which will be transformed to produce the target dataset) and the corresponding gold standard. The produced benchmark (source and target dataset, gold standard) is then sent to an email address (also specified via the interface), allowing the user to test IM systems by comparing the produced matches (between the source and target dataset) against the gold standard. The

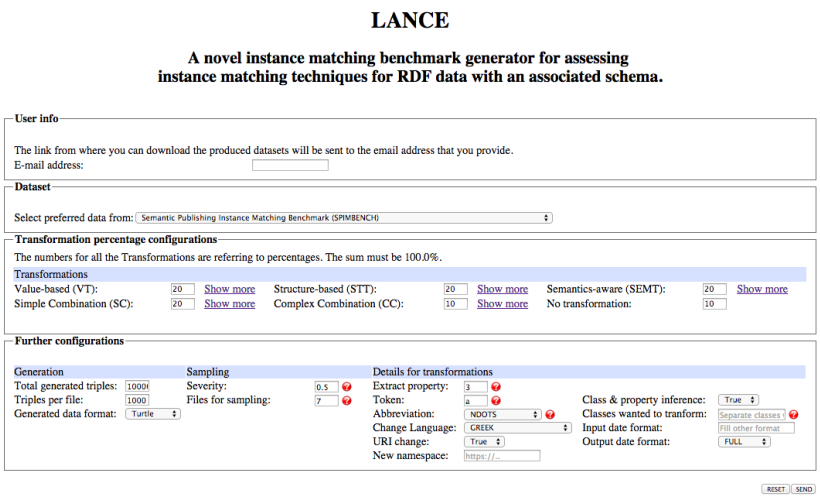

Fig. 2. LANCE demo

benchmark generation is based on a set of configuration parameters which can be tuned via the interface (see Figure 2). The configuration parameters specify the part of the schema and data to consider when producing the target dataset as well as the percentage and type of transformations to consider. The idea behind configuration parameters is to allow one to tune the benchmark generator into producing benchmarks of varying degrees of difficulty which test different aspects of an instance matching tool. The interested reader may also find a video demonstrating the basic functionality in `http://tinyurl.com/ou69jt9`.

References

1. R. Isele, A. Jentzsch, et al. Silk Server - Adding missing Links while consuming Linked Data. In *COLD*, 2010.
2. A.-C. Ngonga Ngomo and Soren Auer. LIMES - A Time-Efficient Approach for Large-Scale Link Discovery on the Web of Data. *IJCAI*, 2011.
3. K. Stefanidis, V. Efthymiou, et al. Entity resolution in the web of data. In *WWW, Companion Volume*, 2014.
4. Ontology Alignment Evaluation Initiative. `http://oaei.ontologymatching.org/`.
5. K. Zaiss, S. Conrad, et al. A Benchmark for Testing Instance-Based Ontology Matching Methods. In *KMIS*, 2010.
6. B. Alexe, W.-C Tan, et al. STBenchmark: Towards a benchmark for mapping systems. In *PVLDB*, 2008.
7. T. Saveta, E. Daskalaki, et al. Pushing the Limits of Instance Matching Systems: A Semantics-Aware Benchmark for Linked Data. In *WWW (Companion Volume)*, 2015.
8. T. Saveta, E. Daskalaki, et al. LANCE: Piercing to the Heart of Instance Matching Tools. In *ISWC*, 2015. To appear.