Web Semantics: Science, Services and Agents on the World Wide Web 39 (2016) 1–14

Contents lists available at ScienceDirect

Web Semantics: Science, Services and Agents on the World Wide Web

journal homepage: www.elsevier.com/locate/websem

Instance matching benchmarks in the era of Linked Data

Evangelia Daskalaki *, Giorgos Flouris, Irini Fundulaki, Tzanina Saveta

FORTH-ICS, Greece

ARTICLE INFO

Article history:
Received 10 November 2015
Received in revised form
15 June 2016
Accepted 22 June 2016
Available online 1 July 2016

Keywords:
Benchmarking
Instance matching
Ontology matching
Semantic Web Data
Ontologies
Linked Data

ABSTRACT

The goal of this survey is to present the state of the art instance matching benchmarks for Linked Data. We introduce the principles of benchmark design for instance matching systems, discuss the dimensions and characteristics of an instance matching benchmark, provide a comprehensive overview of existing benchmarks, as well as benchmark generators, discuss their advantages and disadvantages, as well as the research directions that should be exploited for the creation of novel benchmarks, to answer the needs of the Linked Data paradigm.

1. Introduction

The number of datasets published in the Web of Data as part of the Linked Data Cloud is constantly increasing. The Linked Data paradigm is based on the unconstrained publication of information by different publishers, and the interlinking of Web resources; the latter includes "same-as" interlinking, i.e., the identification of resources described in different datasets that correspond to the same real-world entity. In most cases, the latter type of identification is not explicit in the dataset and must be automatically determined using *instance matching* tools (also known as record linkage [1], duplicate detection [2], entity resolution [3,4,75,76], deduplication [5], merge-purge [6], entity-identification [7], object identification [8], and data fusion [9]).

For example, searching into the Geonames[1] dataset for the resource "Athens" would return the city of Athens in Greece, accompanied with a map of the area and information about the place. Additional information about the same place can be found also in other datasets, for instance in DBpedia[2]; exploiting both information sources requires the identification that these two different web resources (coming from different datasets) correspond to the same real-world entity.

There are various reasons why the same real-world entity is described in different sources. For instance, as mentioned above, in open and social data, anyone is an autonomous data publicist, and simply chooses his preferred representation or the one that best fits his application. Further differences may be due to different data acquisition approaches such as the processing of scientific data. In addition, entities may evolve and change over time, and sources need to keep track of these developments, which is often either not possible or very difficult (especially when this happens in a synchronous way). Finally, when integrating data from multiple sources, the process itself may add (new) erroneous data. Clearly, these reasons are not limited to problems that did arise in the era of Web of Data, it is thus not surprising that instance matching systems have been around for several years [2,10].

The large variety of instance matching techniques requires their comparative evaluation to determine which one is best suited for a given context. Performing such an assessment generally requires well defined and widely accepted benchmarks to determine the weak and strong points of the proposed techniques and/or tools. Furthermore, such benchmarks typically motivate the development of more performant systems in order to overcome identified weak points. Therefore, well-defined benchmarks help push the limits of existing systems, advancing both research and technology.

A benchmark is, generally speaking, a set of tests against which the performance (quality of output, efficiency, effectiveness) of a system is measured.

This survey aims to assess the current state of the art instance matching benchmarks for Linked Data. In particular, we start

* Corresponding author.
E-mail addresses: eva@ics.forth.gr (E. Daskalaki), fgeo@ics.forth.gr (G. Flouris), fundul@ics.forth.gr (I. Fundulaki), jsaveta@ics.forth.gr (T. Saveta).

[1] Geonames http://www.geonames.org/.

[2] DBpedia http://dbpedia.org/.

http://dx.doi.org/10.1016/j.websem.2016.06.002
1570-8268/

by explaining why we choose to study the problem of instance matching benchmarks for Linked Data (Section 2). In Section 3, we describe the characteristics, objectives and main components of an instance matching benchmark. Then, we present benchmark generators for Linked Data (Section 4). In Section 5, we analyse the most important instance matching benchmarks that have been proposed in the literature; the presentation gives particular focus in comparing the characteristics of the different benchmarks and explaining their advantages and drawbacks. This analysis is used in Section 6 to provide guidelines for selecting the proper benchmark for the different contexts and to propose interesting types of benchmarks that could be developed in the future.

Given the increasing importance of instance matching for Linked Data and the plethora of available tools for the task, we believe that a survey on benchmarks for such tools is timely in order to raise awareness on the different existing instance matching evaluation methodologies. To the best of our knowledge, this is the first survey of benchmarks for instance matching tools for Linked Data.

2. Setting the scope

The instance matching problem has been considered for more than half a decade in Computer Science [11] and has been mostly considered for relational data. There has been significant work on instance matching techniques for relational data [1,12,13]. In this context, the problem is well defined: the data is well structured and the focus of the approaches was on discovering differences between values of relation attributes (i.e., value variations). Consequently, the proposed solutions did not have to focus on variations in structure or semantics but simply focus on value variations. In addition, the data processed by the proposed algorithms is dense and usually originated from a very limited number of, well curated, sources.

The first approaches for instance matching for general Web of Data addressed the problem for XML data [14]. In principal, XML data may exhibit strong structural variations (as no schema is necessarily imposed), however, solutions proposed for XML have typically assumed that the data conform to the same schema (i.e., data from different schemata need to be mapped to a common schema before performing instance matching) [14]. Thus, the structural variations between instances are limited to the instance level (e.g., number of occurrences, optional elements, etc.) and not at the schema level. Finally, the proposed methods focus on data that are typically dense.

In the era of Linked Data the picture is different. Linked Data are described by expressive schemas that carry rich semantics expressed in terms of the RDF Schema Language (RDFS) and the OWL Web Ontology Language. RDFS and OWL vocabularies are used by nearly all data sources in the LOD[3] cloud. According to a recent study,[4] 36.49% of LOD use various fragments of OWL so it is imperative that we consider the constraints expressed in such schemas when developing instance matching tools and benchmarks. Consequently, the variations in the huge number of data sources are value, structural as well as logical [15]. As far as semantics are concerned, when paired with a suitable reasoning engine, Linked Data allow implicit relationships to be inferred from the data [15], which was not possible with relational data and XML data.

Due to these reasons, instance matching systems that have been designed for relational or XML data cannot fully exploit the aforementioned heterogeneities and thus failed to deliver good matching results.

Furthermore, according to [9], there exist specific requirements that distinguish the Linked Data from other instance matching workloads, which arise from the autonomy of data sources and the uncertainty of quality-related meta-information. Thus, it is required to assess data quality in order to resolve inconsistencies.

This survey aims at *describing the current state of the art in instance matching benchmarks, with particular focus on the case of Linked Data*, which, as explained above, present differences in the nature of the data (values and structure), but also in the semantic load they carry.

3. Instance matching benchmarks

A benchmark is a set of tests against which the performance (quality of output, efficiency, effectiveness) of a system is measured. Benchmarking, from a philosophical point of view, is "the practice of being humble enough to admit that someone else is better at something and wise enough to try to learn how to match and even surpass them at it" [16]. The underlying meaning of the above quotation is that it is certainly not easy to be the best, but what matters most is trying to become the best and this can only be done through assessment and identification of weak points, which can be worked upon and improved. So, benchmarking aims at providing an objective basis for assessments. In this way, benchmarks help computer systems to compare, to assess their performances, and last but not least, to push systems to get further. Due to the fact that the performance of the systems varies enormously from one application domain to another [17], there does not exist a single benchmark that can measure the performance of computer systems on all contexts. Thus, it is essential to have domain-specific benchmarks that specify a typical workload for the corresponding domain; this survey focuses on IM benchmarks for Linked Data, for the reasons that have been explained in Section 2.

The results of various systems on a benchmark gives a rough estimate of their performance. However, such estimates are always relative, i.e., in relation to the results of other systems for the same benchmark [17]. Along these lines, if a system shows good results for one specific benchmark, we could conclude that the system can handle very well the workload of the benchmark, but we cannot easily come to a conclusion about the benchmark itself (e.g., how well it addresses the challenges it sets). If though the majority of systems provide good results, we can say that the benchmark addresses trivial cases.

In order for the systems to be able to use the benchmarks and report reliable results, the benchmarks must have specific characteristics. First of all, they have to be *open and accessible* for all interested parties, so that the results can be compared to each other. Open means that they are free to use and accessible means that they are easily available to the interested parties.

Moreover, benchmarks also have to be *persistent*. By this we mean that the components of one benchmark should not evolve or change through time, so as to make the results of different systems (obtained at different times) comparable.

Note that this requirement rules out testing datasets (sometimes called "benchmarks") which are based on datasets obtained from the Web (without being versioned). For example in [18–20], and [21] authors use very well-known datasets like LinkedMDB [22] and DBpedia to evaluate their well-known systems, but since these datasets evolve through time, it is very difficult to reproduce the exact same test-sets, and thus run the same tests again. Due to these reasons these testing datasets are not considered as benchmarks.

[3] https://www.w3.org/DesignIssues/LinkedData.html.
[4] http://linkeddatacatalog.dws.informatik.uni-mannheim.de/state/.