Data & Knowledge Engineering 91 (2014) 34–51

Contents lists available at ScienceDirect

Data & Knowledge Engineering

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

Automatically generating data linkages using class-based discriminative properties

Wei Hu [a,b,*], Rui Yang [b], Yuzhong Qu [a,b]

[a] State Key Laboratory for Novel Software Technology, Nanjing University, China
[b] Department of Computer Science and Technology, Nanjing University, China

ARTICLE INFO

Article history:
Received 17 June 2013
Received in revised form 1 March 2014
Accepted 4 March 2014
Available online 27 March 2014

Keywords:
Data linkage
Instance matching
Object coreference resolution
Semantic Web
Discriminability
Ontologies

ABSTRACT

A challenge for Linked Data is to link instances from different data sources that denote the same real-world object. Millions of high-quality `owl:sameAs` linkages have been generated, but potential ones are still considerable. Traditional similarity-based methods to this data linkage problem do not scale well since they exhaustively compare every pair of instances. In this paper, we propose an automatic approach to data linkage generation for Linked Data. Specifically, a highly-accurate training set is automatically generated based on equivalence reasoning and common prefix blocking. The contexts of the instances in the training set, after extracting, are pairwise matched in order to learn discriminative property pairs supporting linkage discovery. For a particular class pair and a pay-level-domain pair, the discriminability of each property pair is measured, and a few property pairs with high discriminability are aggregated in order to be reused in the future to link instances between the same classes and domains. The experimental results show that our approach achieves good accuracy against some complex methods in two OAEI tests and the BTC2011 dataset.

1. Introduction

The Semantic Web (SW) is an effort by the W3C Semantic Web Activity, with the purpose of realizing data integration and sharing among different applications and parties. As of today, many prominent ontologies have been developed for data publishing in various domains, which suggest common classes and properties widely used across data sources.

At the instance level, however, there is lack of agreement among sources on the use of common URIs to denote a real-world object. Due to the distributed nature of the SW, it frequently happens that multiple instances in diverse sources denote the same object, i.e., refer to an identical thing (also known as URI aliases [1] or coreferents). Such examples exist in the areas of personal profiles, academic publications, media or geographical data, etc.

Data linkage, also referred to as instance matching or object coreference resolution, aims at linking different instances for the same object. It is important to data-centric applications such as heterogeneous data integration or mining systems, SW search engines and browsers. Driven by the Linking Open Data (LOD) initiative, millions of instances have been linked with `owl:sameAs` explicitly [2], whose semantics defines that all the URIs linked with this property should identify the same resource. But compared to billions of URIs on the SW, there still exists a large amount of instances that potentially denote the same objects without being interlinked yet. For example, at least 70 instances crawled by the Falcons search engine [3] seem to denote Tim Berners-Lee, the director of W3C, but merely six have been linked with `owl:sameAs`. An analysis on the LOD cloud also indicates that, out of 31 billion RDF statements less than 500 million represent linkages between data sources, and most sources only link to one another.[1]

* Corresponding author at: State Key Laboratory for Novel Software Technology, Nanjing University, China. Tel./fax: +86 25 8968 0923.
E-mail addresses: whu@nju.edu.cn (W. Hu), ryang@smail.nju.edu.cn (R. Yang), yzqu@nju.edu.cn (Y. Qu).

[1] http://lod-cloud.net/state/ (09/19/2011).

http://dx.doi.org/10.1016/j.datak.2014.03.001
0169-023X/© 2014 Elsevier B.V. All rights reserved.

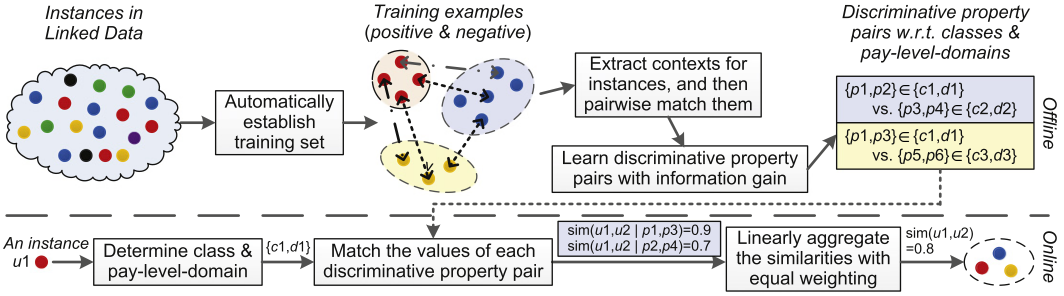
Instances in Linked Data
Automatically establish training set
Training examples (positive & negative)
Extract contexts for instances, and then pairwise match them
Learn discriminative property pairs with information gain
Discriminative property pairs w.r.t. classes & pay-level-domains
{p1,p2}∈{c1,d1} vs. {p3,p4}∈{c2,d2}
{p1,p3}∈{c1,d1} vs. {p5,p6}∈{c3,d3}
Offline
An instance u1
Determine class & pay-level-domain
{c1,d1}
Match the values of each discriminative property pair
sim(u1,u2 | p1,p3)=0.9
sim(u1,u2 | p2,p4)=0.7
Linearly aggregate the similarities with equal weighting
sim(u1,u2) =0.8
Online

Fig. 1. Overview of the approach.

In the SW community, traditional work addresses the data linkage problem mainly from two directions: one is by performing *equivalence reasoning* in terms of standard OWL semantics, e.g., through `owl:sameAs` and some "key" properties [4,5]; the other is by *similarity computation*, with the intuition that instances denote the same object if having similar property–values [6,7]. Recent work also uses machine learning and crowdsourcing to cope with complex data linkage tasks [8–11]. Generally speaking, the reasoning-based methods infer explicit linkages but may miss many potentials, while the similarity-based ones often suffer from high computational costs as they exhaustively compare all pairs of instances [12]; many of them have not been aware of the commonalities behind the abstract types of the instances and their publishers. For example, different data publishers prefer social security number, login name, address or even their combinations to disambiguate customers, but hobby or age is less likely to be used. It will facilitate data linkage in the future if such properties can be learnt and reused.

In this paper, we propose an automatic approach, called *ADL*, which differs from current similarity-based methods in learning a set of important properties for disambiguating instances (referred to as *discriminative properties*). The methodological steps of ADL, shown in Fig. 1, can be divided into the offline part and the online part:

- For the offline learning, a highly-accurate *training set* is automatically established. The training set consists of two sets of instance pairs holding the linkages or not, referred to as *positive examples* and *negative examples*, resp. The *contexts* (i.e., a kind of integrated units over RDF triples) for the instances in the training set are extracted in terms of RDF sentences [13], and pairwise matched with a lightweight linguistic matcher V-Doc [14], in order to discover discriminative property pairs, where a discriminative property pair consists of two matchable properties discriminative to link instances. For a specific class pair and a pay-level-domain pair,[2] the discriminability of each property pair is measured by information gain, revealing the global and implicit preference of data publishers on characterizing a type of objects.
- For the online linking, given a new instance as input, the class that it belongs to and its pay-level-domain are firstly extracted, and then the counterparting classes and pay-level-domains in the training set are chosen. The instances, with the properties in the related discriminative property pairs, are found out, and their values are matched with that of the input using V-Doc. The similarities from different discriminative property pairs are linearly aggregated with equal weighting, in order to determine whether to generate an instance linkage.

We develop an open source tool and test its accuracy on three cases: the PR and NYT tests in the Ontology Alignment Evaluation Initiative (OAEI) as well as the Billion Triples Challenge (BTC2011) dataset. The experimental results show that, compared with several existing methods, our method achieves good precision and recall with the help of only a few discriminative property pairs. Moreover, the proposed approach is ready to be integrated with other methods, e.g., the found discriminative properties can be used for cost-effective candidate selection [12].

This paper is organized as follows. We define the data linkage problem in Section 2 and discuss related work in Section 3. In Sections 4 and 5, we present our approach to learn class-based discriminative property pairs. Evaluation is reported in Section 6. Finally, Section 7 concludes the paper with future work.

2. Problem statement

Let **I** be the set of URIs, **B** be the set of blank nodes and **L** be the set of literals. A triple $\langle s, p, o\rangle \in (\mathbf{I} \cup \mathbf{B}) \times \mathbf{I} \times (\mathbf{I} \cup \mathbf{B} \cup \mathrm{L})$ is called an *RDF triple*. An *RDF graph* $\mathcal{G}$ is a set of RDF triples, and can be serialized to an RDF document.

For an RDF graph $\mathcal{G}$, a URI u is a *class* (resp. *property*) if $\mathcal{G}$ entails the RDF triple $\langle u, \mathrm{rdf:type}, \mathrm{rdfs:Class}\rangle$ (resp. $\langle u, \mathrm{rdf:type}, \mathrm{rdf:Property}\rangle$). If a URI u is not either a class, a property or both, then u is treated as an *instance*, implying the assumption that classes

[2] The *pay-level-domain* is a sub-domain of a public top-level-domain, for which users usually pay, e.g., the pay-level-domain for `www.example.com` is `example.com`. Pay-level-domains allow to identify a realm, where a data publisher is likely to be in control [2].