Web Semantics: Science, Services and Agents on the World Wide Web 23 (2013) 2–15

Contents lists available at ScienceDirect

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

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

Active learning of expressive linkage rules using genetic programming

Robert Isele *, Christian Bizer

Research Group Data and Web Science, University of Mannheim, B6 26, 68131 Mannheim, Germany

ARTICLE INFO

Article history:
Received 14 June 2012
Received in revised form
9 March 2013
Accepted 11 June 2013
Available online 1 July 2013

Keywords:
Entity matching
Duplicate detection
Active learning
Genetic programming
Linkage rules
ActiveGenLink

ABSTRACT

A central problem in the context of the Web of Linked Data as well as in data integration in general is to identify entities in different data sources that describe the same real-world object. Many existing methods for matching entities rely on explicit linkage rules, which specify the conditions which must hold true for two entities in order to be interlinked. As writing good linkage rules by hand is a non-trivial problem, the burden to generate links between data sources is still high. In order to reduce the effort and expertise required to write linkage rules, we present the ActiveGenLink algorithm which combines genetic programming and active learning to generate expressive linkage rules interactively. The ActiveGenLink algorithm automates the generation of linkage rules and only requires the user to confirm or decline a number of link candidates. ActiveGenLink uses a query strategy which minimizes user involvement by selecting link candidates which yield a high information gain. Our evaluation shows that ActiveGenLink is capable of generating high quality linkage rules based on labeling a small number of candidate links and that our query strategy for selecting the link candidates outperforms the query-by-vote-entropy baseline.

1. Introduction

The goal of the Linked Data movement is to extend the Web with a global data space by making data sets accessible according to a set of best practices and by setting RDF links between data sources [1]. While the amount of data that is accessible as Linked Data has grown significantly over the last years, most data sources are still not sufficiently interlinked. Out of the over 31 billion RDF statements published as Linked Data less than 500 million represent RDF links between data sources [2]. Analysis of the Linking Open Data cloud confirms that it represents a weakly connected graph with most publishers only linking to one other data source [2].

A number of link discovery tools have been developed, which generate RDF links between entities in different data sets that represent the same real-world object. Unfortunately, fully automatic link discovery tools do not achieve a satisfying accuracy on many data sets [3]. For this reason, several semi-automatic link discovery tools – such as Silk [4] or LIMES [5] – have been developed. These tools compare entities in different Linked Data sources based on user-provided linkage rules which specify the conditions that must hold true for two entities in order to be interlinked.

Writing good linkage rules by hand is a non-trivial problem as the rule author needs to have a detailed knowledge about the structure of the data sets: first of all, the author needs to choose discriminative properties of the entities to be interlinked together with a distance measure and an appropriate distance threshold. For data sets which are noisy or use different data formats, the property values need to be normalized by employing data transformations prior to comparison. As comparing entities by a single property usually is not sufficient to decide whether both entities describe the same real-world object, the linkage rule has to aggregate the similarity of multiple property comparisons using appropriate aggregation functions.

We illustrate this point with the example of a data set about movies: even within this simple example the linkage rule author faces a couple of challenges. First of all, a comparison solely by film title fails for cases when movies with the same title actually represent distinct movies that have been released in different years. Therefore, the linkage rule needs to compare, at the very least, the titles of the movies as well as their release dates and combine both similarities with an appropriate aggregation function. As data sets can be noisy (e.g., the release dates might be off by a couple of days), the rule author also needs to choose suitable distance measures together with appropriate distance thresholds. Linkage rules must also cover data heterogeneities. For instance, a data source may contain some person names that are formatted as "<first name><last name>" while others are formatted as "<last name>, <first name>". Finding such heterogeneities and adding the specific data transformations to avoid mismatches are often very tedious. Thus, writing a linkage rule is not only a cumbersome but also a time consuming task.

Supervised learning. A way to reduce this effort is to use supervised learning to generate links from the existing reference links, which contain pairs of entities that have been labeled as matches

* Corresponding author. Tel.: +49 17663645978.
E-mail addresses: mail@robertisele.com (R. Isele), chris@bizer.de (C. Bizer).

1570-8268/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.websem.2013.06.001

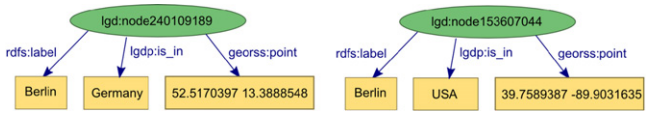

Fig. 1. Example of an entity pair in a geographical data set.

or non-matches. Creating such reference links is much easier than to write linkage rules as it requires no previous knowledge about similarity computation techniques or the specific linkage rule format used by the system. Usually, reference links are created by domain experts who confirm or reject the equivalence of a number of entity pairs from the data sets. For instance, reference links for locations in a geographical data set can be created by labeling pairs of locations as correct or incorrect. Fig. 1 shows an example of an entity pair. In the given example, the pair is to be declined as both entities represent different real-world locations.

Active learning. In order for the supervised learning algorithms to perform well on unknown data, reference links need to include all relevant corner cases. We illustrate this point by having a second look at the example in Fig. 1: while for many cities a comparison by label is sufficient to determine if two entities represent the same real-world city, the given example shows the corner case of distinct cities sharing the same name. If the entity pairs to be labeled by the user are just selected randomly from the data sets, the user has to label a very large number of pairs in order to include these rare corner cases reliably. As manually labeling link candidates is time-consuming, methods to reduce the number of candidates which need to be labeled are desirable.

The fundamental idea of active learning in the context of entity matching is to reduce the number of link candidates which need to be labeled by actively selecting the most informative candidate for being labeled by the user.

Our contribution. In this article, we present ActiveGenLink, an algorithm for learning linkage rules interactively using active learning and genetic programming. ActiveGenLink learns a linkage rule by asking the user to confirm or reject a number of link candidates which are actively selected by the algorithm. Compared to writing linkage rules by hand, ActiveGenLink lowers the required level of expertise as the task of generating linkage rules is automated by the genetic programming algorithm while the user only has to verify a set of link candidates. The employed query strategy for selecting link candidates minimizes user involvement by selecting the links with the highest information gain for manual verification. Within our experiments, ActiveGenLink outperformed state-of-the-art unsupervised approaches after manually labeling a few link candidates (less than 5 within our experiments). In addition, ActiveGenLink is capable of generating linkage rules with a comparable performance than the supervised GenLink algorithm [6] by labeling a much smaller number of link candidates (between 15 and 50 within our experiments). ActiveGenLink chooses which properties to compare, it chooses appropriate distance measures, aggregation functions, and thresholds, as well as data transformations that are applied to normalize data prior to comparison.

This article makes the following contributions:

1. we propose the ActiveGenLink algorithm which applies genetic programming and active learning to the problem of learning linkage rules for generating RDF links in the context of the Web of Data.
2. the learned rules are more expressive than the linkage rules learned in the previous work on active learning of linkage rules as our algorithm combines different similarity measures non-linearly and also determines the data transformations that should be employed to normalize data prior to comparison.
3. we propose a query strategy that minimizes the number of links that need to be labeled by the user and outperforms the query-by-vote-entropy strategy, which has been used in previous work.
4. we have implemented the proposed approach in the Silk Workbench, a web application which can be used by Linked Data publishers and consumers to set RDF links. The Silk Workbench is part of the Silk Link Discovery Framework and is available for download under the terms of the Apache License.

This article builds on our previous work presented at two occasions:

- in [6], we present the GenLink algorithm for learning expressive linkage rules from a set of existing reference links using genetic programming.
- in [7], we present an approach which combines genetic programming and active learning to generate expressive linkage rules interactively.

This article extends the previously presented active learning approach with a novel query strategy, which further minimizes the number of links that need to be labeled by the user as well as a more extensive experimental evaluation. Although in this paper we focus on interlinking data sources in the context of the Web of Linked Data, our approach is not limited to this use case and can be applied to entity matching in other areas – such as in the context of relational databases – as well.

Outline. This article is organized as follows: Section 2 formalizes the entity matching problem. Section 3 introduces our linkage rule representation. Based on that, Section 4 describes the ActiveGenLink workflow. Section 5 describes the genetic programming approach used for learning linkage rules from existing training data. Section 6 describes the proposed query strategy for selecting link candidates in detail. Section 7 introduces our approach of building the initial pool of unlabeled links. Section 8 presents the results of our experimental evaluation. Section 9 discusses related work. Section 10 presents the implementation of ActiveGenLink in the Silk Workbench.

2. Problem definition

We consider the problem of matching entities between two data sources A and B. The objective is to determine which entities in A and B identify the same real-world object.

The general problem of entity matching can be formalized as follows [8]:

Definition 1 (*Entity Matching*)**.** Given two data sources A and B, find the subset of all pairs of entities for which a relation $\sim_R$ holds:

$$M = \{(a, b);\ a \sim_R b, a \in A, b \in B\}.$$

Similarly, we define the set of all pairs for which $\sim_R$ does not hold:

$$U = (A \times B) \setminus M.$$

The purpose of relation $\sim_R$ is to relate all entities which represent the same real-world object.

In some cases a subset of M and U is already known prior to matching. Such *reference links* can, for instance, originate from previous data integration efforts. Alternatively, they can be created by domain experts who simply need to confirm or reject the equivalence of entity pairs from the data sets.

Definition 2 (*Reference Links*)**.** A set of positive reference links $R_+ \subseteq M$ contains pairs of entities for which relation $\sim_R$ is known to hold (i.e. which identify the same real-world object). Analogously, a set of negative reference links $R_- \subseteq U$ contains pairs of entities for which relation $\sim_R$ is known to not hold (i.e. which identify different real-world objects).