Enhancing Dataset Quality Using Keys

Tommaso Soru, Edgard Marx, and Axel-Cyrille Ngonga Ngomo
{tsoru,marx,ngonga}@informatik.uni-leipzig.de

AKSW, Department of Computer Science, University of Leipzig

Abstract. The Linked Data principles provide a decentral approach for publishing structured data in RDF on the Web. A consequence of this architectural choice is a high variance in the quality of the RDF datasets which constitute the Linked Data cloud. In this demo paper, we address a particular aspect of quality, i.e., the discriminability of resources. During our demo, we will present our simple three-step approach and interface, which allows data publishers to detect the resources in their dataset that are indistinguishable with respect to a given set of properties. Our approach is highly scalable as it relies on ROCKER, a novel algorithm for key discovery. Our evaluation on DBpedia suggests that even very commonly-used data sources are still in need to significant improvement to abide by the discriminability criterion.

1 Introduction

The quality of RDF datasets on the Web varies significantly [5]. While several approaches have been developed to check the quality of datasets (see [2,5] for an overview), the discriminability of resources (see Section 2 for a formal definition) has been paid little attention to. However, improving the discriminability of resources has been shown to be beneficiary for the quality of the links created by automatic linking processes [4]. Hence, this process promises datasets to improve their abiding by the fourth Linked Data principle.

In this demo, we present a tool that addresses exactly this gap. Our framework[1] makes use of ROCKER [3], a state-of-the-art algorithm for key discovery, to detect indistinguishable resources and facilitate the curation of these resources. The following section introduces some preliminaries; we then describe the approach implemented on the demo and illustrate a use case; thereafter, we conclude.

2 Preliminaries

Let K be a finite RDF knowledge base containing instances which belong to a given class and their Concise Bounded Description (CBD).[2] K can be regarded as a set of triples $(s, p, o) \in (\mathcal{R} \cup \mathcal{B}) \times \mathcal{P} \times (\mathcal{R} \cup \mathcal{L} \cup \mathcal{B})$, where $\mathcal{R}$ is the set of all resources, $\mathcal{B}$ is the set of all blank nodes, $\mathcal{P}$ the set of all predicates and $\mathcal{L}$ the set of all literals. We call

[1] Available online at http://rocker.aksw.org/.
[2] For the definition of CBD, see http://www.w3.org/Submission/CBD/.

two resources $r_1, r_2 \in \mathcal{R}$ *distinguishable* w.r.t. a set of properties $P = \{p_1, \dots p_n\}$ iff $\exists p \in \{p_1, \dots p_n\} \exists o : ((r_1, p, o) \wedge \neg(r_2, p, o)) \vee (\neg(r_1, p, o) \wedge (r_2, p, o))$.

Given a knowledge base K, the idea behind *key discovery* is to find one or all sets of properties which make their respective subjects distinguishable in K. We call a set of properties $P \subseteq \mathcal{P}$ a *key* for a knowledge base K (short: key, denoted $key(P, K)$) if all resources in K are distinguishable w.r.t. P. Let us consider the smallest set $S' \subseteq S$ such that every pair of distinct elements in $S \setminus S'$ are distinguishable from each other. The discriminability score of S w.r.t. P is then calculated as $score_P(S) = \frac{|S \setminus S'|}{|S|}$. P is called a *k-almost-key* if S' has cardinality $\leq k$.

3 Enhancing Dataset Quality Using Keys

The intuition behind our framework is based on the following principle. Since keys are defined as *unique descriptions of resources*, any description collision can be considered as a potential error. We then assume that the error rate for a key is no lower than a threshold value α. Thereafter, we ask ROCKER to find any P such that $score_P(S) \geq \alpha = 1 - \frac{k}{|S|}$. We expect that for each k-almost-key P, curating the k resources in S' would lead to a dataset with higher accuracy, easier to link and thus fitter for use in applications which rely, e.g., on federated data sources.

A similar approach was introduced in [1], where the authors discover `owl:sameAs` links among resources which do not obey by almost-keys[3]. However, the definition of key adopted in the paper presented some imperfections, as discussed in [3].

The contributions of our work can be listed as follows: (i) We devise a method for the detection of data issues using keys; (ii) To the best of our knowledge, we implement the first interface for dataset repair using keys; (iii) We release a vocabulary[4] for key discovery which relies on a more correct definition of keys; (iv) We provide a RESTful API endpoint for our algorithm ROCKER.

Our framework implements a three-step approach composed by *threshold selection*, *key selection*, and *issue visualization and export*. During the demo, we will show all of the steps presented below:

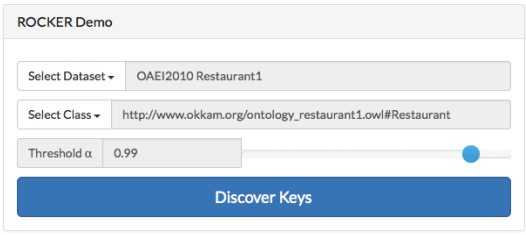
Threshold Selection. The first step, i.e. the threshold selection, is depicted in Figure 1a. As can be seen, users choose the dataset and the class on which the key discovery will be carried out. A threshold value of discriminability is also required. This parameter represents the discriminability factor, i.e. the minimum score for a set of properties to be considered an almost-key. Previous research has shown that there is no direct correlation among α and the amount of retrieved almost-keys, nor the resources (i.e., runtime and memory) needed for the computation [3].

Key Selection. The key selection step is depicted in Figure 1c. In this step, the result of the key discovery task is shown to the user. Each composite almost-key can be selected to show the respective resources with issues, if any.

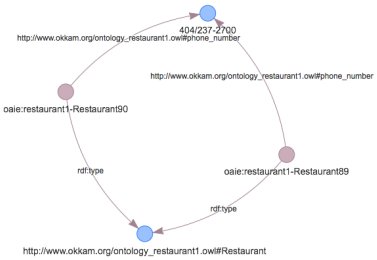
[3] Henceforth, we will call them *almost-keys* to generalize for any *k-almost-key*.

[4] Described at `http://rocker.aksw.org/vocab.xhtml`.

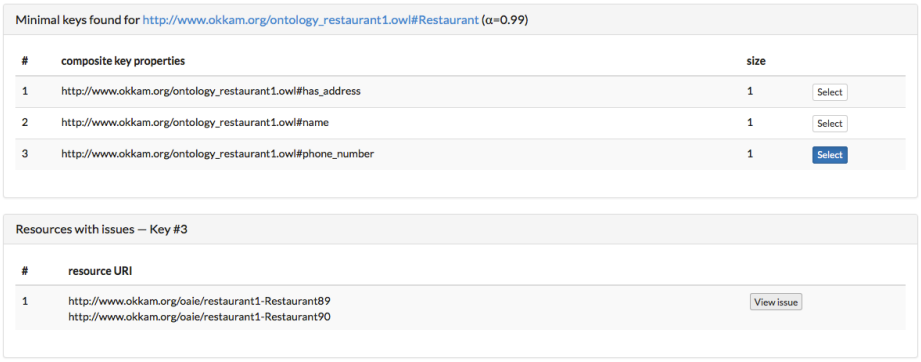
Fig. 1: Enhancing dataset quality using ROCKER.



(a) First step, threshold selection.



(b) Third step, issue visualization.



(c) Second step, key selection.

Issue Visualization and Export. In the last step, users are shown the subgraph containing the resources having issues (red circles), their types and common object values (blue circles). Figure 1b shows how the subgraph is rendered to the user. Finally, almost-keys and resources with issues can be exported in a human- and machine-readable format.

4 Use Case

In this demo, we show how our approach can be applied on one real-world and two synthetic datasets. The first dataset was generated from DBpedia 3.9[5] and contains 360 instances of class `dbpedia-owl:Monument` and their CBD. The remaining two datasets belong to the 2010 Instance Matching track of the Ontology Alignment Evaluation Initiative[6]. The former describes persons, while the latter describes restaurants. All datasets are available online on the respective websites.

In order to understand the workflow, we now illustrate a use case in detail, which will be shown during the demo session. Figure 1 shows the three stages applied to

[5] `http://oldwiki.dbpedia.org/Downloads39/`
[6] `http://oaei.ontologymatching.org/2010/im/`

our use case, which refers to the *OAEI Restaurant1* dataset. As can be seen, class `oaei:Restaurant` and a threshold value of $\alpha = 0.99$ are selected. As soon as the user presses the `Discover Keys` button, ROCKER is launched via a RESTful API call, whereas a loading screen is presented to the user. In the following step, the user is asked to select almost-keys and issues. Since the first two almost-keys `oaei:has_address` and `oaei:name` are perfect keys (i.e., they achieve a score of 1), they present no issues. On the other hand, the third key – composed only by property `oaei:phone_number` – presents one issue, which is then selected by the user.

Please note that some issues may be critical only for certain almost-keys, depending on the specific domain. While common sense would suggest that two restaurants having the same address or name may coexist, two restaurants having the same phone number are unlikely to find. Therefore, resources related to some almost-keys – `oaei:phone_number` in our example – have more probability to contain errors than others.

The subgraph displaying the issue is then rendered. In the graph, all values for each property (in addition to `rdf:type`) are shown. Should the graph display no object values, the resources are not distinguishable by the fact of sharing only *null* values. Here, two restaurants (displayed in red color) share the same `oaei:phone_number`, therefore they are not distinguishable w.r.t. the almost-key $\{$`oaei:phone_number`$\}$. The user can thus export the data to a single file. In case the user thinks that the two restaurants refer to the same real-world entity, then an `owl:sameAs` link should subsist among them. Otherwise, one of the two phone numbers is probably incorrect.

5 Summary

This demo paper presents a method to enhance the quality of Linked Datasets using keys. The web interface shows how ROCKER, a state-of-the-art algorithm for key discovery, can help finding inaccuracies in the data. Furthermore, we provide a RESTful API endpoint for key discovery, yielding results in JSON-LD format. In future work, we expect to open the possibility for users to upload their own datasets and we will provide an online editor in order to speed up the data repair task.

References

1. M. Atencia, J. David, and F. Scharffe. Keys and pseudo-keys detection for web datasets cleansing and interlinking. In *Knowledge Engineering and Knowledge Management*. 2012.
2. D. Kontokostas, P. Westphal, S. Auer, S. Hellmann, J. Lehmann, R. Cornelissen, and A. Zaveri. Test-driven evaluation of linked data quality. In *Proceedings of the 23rd International Conference on World Wide Web*, 2014.
3. T. Soru, E. Marx, and A.-C. Ngonga Ngomo. ROCKER: A refinement operator for key discovery. In *Proceedings of the 24th International Conference on World Wide Web*, 2015.
4. D. Symeonidou, V. Armant, N. Pernelle, and F. Saïs. SAKey: Scalable almost key discovery in RDF data. In *International Semantic Web Conference*, 2014.
5. A. Zaveri, A. Rula, A. Maurino, R. Pietrobon, J. Lehmann, and S. Auer. Quality assessment for linked data: A survey. *Semantic Web Journal*, 2015.