Discovering Keys in RDF/OWL Datasets with KD2R *

Danai Symeonidou
LRI, Université Paris-Sud
PCRI, Bat 650
91405 Orsay, France
danai.symeonidou@lri.fr

Nathalie Pernelle
LRI, Université Paris-Sud
PCRI, Bat 650
91405 Orsay, France
nathalie.pernelle@lri.fr

Fatiha Saïs
LRI, Université Paris-Sud
PCRI, Bat 650
91405 Orsay, France
fatiha.sais@lri.fr

ABSTRACT

KD2R allows the automatic discovery of composite key constraints in RDF data sources that conform to a given ontology. We consider data sources for which the Unique Name Assumption is fulfilled. KD2R allows this discovery without having to scan all the data. Indeed, the proposed system looks for maximal non keys and derives minimal keys from this set of non keys. KD2R has been tested on several datasets available on the web of data and it has obtained promising results when the discovered keys are used to link data. In the demo, we will show the functionality of our tool and we will show on several datasets that the keys can be used in a data linking tool.

1. INTRODUCTION

Establishing identity links between data items allows crawlers, browsers and applications to combine information from different RDF data sources. Many approaches aim to detect sameAs links between data items (see [3] for a survey). Most of these approaches use either specific expert rules that specify conditions that two data items must fulfill in order to be linked [5, 9, 1] or keys that are declared in the ontology [7]. Nevertheless, when the data are numerous and heterogeneous, these rules or keys cannot easily be specified by a human expert. In [4, 6] linkage rules are learnt on a set of reference links while some other approaches focus on key discovery [2, 8].

We present an extension of KD2R [8], an automatic tool for key discovery in RDF data sources that conform to OWL ontologies. This tool aims to discover key constraints that are composed of several properties. Indeed, non composite keys (e.g. ISBN for books) are rare. KD2R discovers keys from datasets where different URIs refer to different world entities (i.e. Unique Name Assumption). Since we work under the Open World Assumption (OWA), KD2R uses heuristics to interpret the absence of information. To avoid scanning all the data, KD2R discovers first maximal non keys before inferring the keys. Indeed, finding two instances that share the same values for the considered set of property expressions suffice to be sure that this set is a non key. Furthermore, it exploits key inheritance between classes in order to prune the non key search space. KD2R tool has been evaluated on different data sets. In particular, we can show that when a linking tool exploits these keys that are discovered automatically, relevant identity links can be generated.

*This work has been supported by the French National Research Agency (ANR) in the setting of the Qualinca project.

2. KD2R SYSTEM

Since RDF data sources might contain descriptions that are incomplete (Open World Assumption), it is not meaningful to discover keys from a RDF datasets. Heuristics are needed to declare that a key is valid for a dataset.

2.1 Optimistic and Pessimistic heuristics

We consider that a set of property expressions is a *key* for a class in a data source if for all pairs of distinct instances of this class, there exists a property expression in this set such that all the values are pairwise distinct (objects or literal values). We consider that a set of property expressions is a *non key* for a class if there exist two distinct instances of this class that share at least one value for all the property expressions of this set. Some combinations of property expressions are neither keys nor non keys: a set of property expressions is called an *undetermined key* for a class if it is not a non key and there exist two instances of the class such that the instances share the same values for a subset of the property expressions, and the remaining property expressions are unknown for at least one of the two instances.

Distinguishing undetermined keys from keys and non keys allows us to use them differently. Using a *pessimistic heuristic*, the property for which no value is given can take all the values that appear in the data source. Therefore, the undetermined keys will not be considered as keys. Using an *optimistic heuristic*, we consider that the not given property values are different from all the values that appear in the data source for this property. This leads to consider the undetermined keys as keys.

2.2 KD2R main steps

In Figure 1 we show the main steps of KD2R system. The user gives an ontology and a set of data sources. Then, the tool uses the pessimistic or optimistic heuristic to discover the key constraints for each RDF data source independently.

In each data source, KD2R is applied on the classes in topologically sorted order. This way, the keys that are discovered in the super-classes are exploited in the processing of their sub-classes. For a given data source s_i and a given class c we apply Key-Finder algorithm which aims at finding keys for the class c that are valid in the data source s_i. Key-Finder starts by building a prefix tree for this class to represent its instances (see Figure 1). Using this representation the sets of maximal undetermined keys and maximal non keys are computed. These sets of undetermined keys and non keys, are used to derive the set of minimal keys. The obtained keys are then merged in order to compute the set of key constraints that are valid for both data sources (see Figure 2).

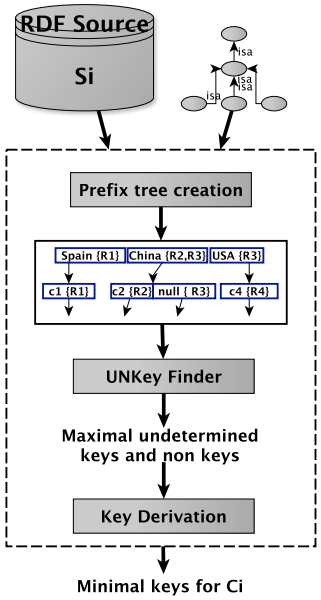

Figure 1: Key discovery for one data source

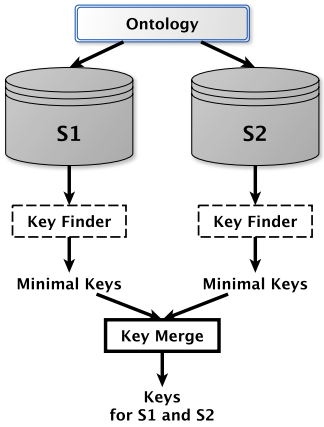

Figure 2: Key merge for two data sources

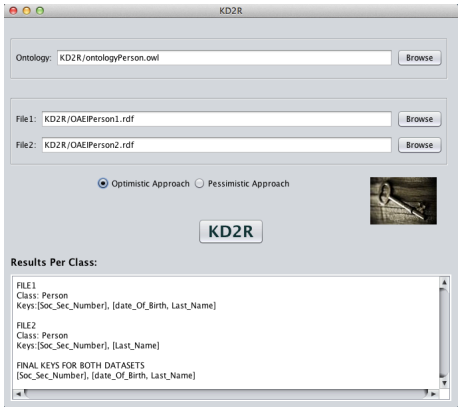

Figure 3: KD2R Graphical User Interface

3. DEMONSTRATION

We will use several datasets to show the keys that can be found using pessimistic or optimistic heuristics. Then, the data linking tool called N2R [7] will be applied on datasets for which a gold standard is available. Thus we will show and compare results that can be odtained when no keys are available (all the properties are used with the same importance and agregated by an average function), when expert keys are available and when KD2R keys are available.

4. REFERENCES

[1] A. Arasu, C. Ré, and D. Suciu. Large-scale deduplication with constraints using dedupalog. In *ICDE*, pages 952–963, 2009.

[2] M. Atencia, J. David, and F. Scharffe. Keys and pseudo-keys detection for web datasets cleansing and interlinking. In *EKAW*, pages 144–153, 2012.

[3] A. Ferrara, A. Nikolov, and F. Scharffe. Data linking for the semantic web. *Int. J. Semantic Web Inf. Syst.*, 7(3):46–76, 2011.

[4] R. Isele and C. Bizer. Learning expressive linkage rules using genetic programming. *PVLDB*, 5(11):1638–1649, 2012.

[5] W. L. Low, M. L. Lee, and T. W. Ling. A knowledge-based approach for duplicate elimination in data cleaning. *Information Systems*, 26:585–606, December 2001.

[6] A.-C. N. Ngomo and K. Lyko. Eagle: Efficient active learning of link specifications using genetic programming. In *9th Extended Semantic Web Conference (ESWC)*, pages 149–163, 2012.

[7] F. Saïs, N. Pernelle, and M.-C. Rousset. Combining a logical and a numerical method for data reconciliation. *Journal on Data Semantics*, 12:66–94, 2009.

[8] D. Symeonidou, N. Pernelle, and F. Saïs. Kd2r: A key discovery method for semantic reference reconciliation. In *OTM Workshops*, pages 392–401, 2011.

[9] J. Volz, C. Bizer, M. Gaedke, and G. Kobilarov. Discovering and maintaining links on the web of data. In *Proceedings of the 8th International Semantic Web Conference*, ISWC '09, pages 650–665, Berlin, Heidelberg, 2009. Springer-Verlag.