KeywDB: A System for Keyword-Driven Ontology-to-RDB Mapping Construction

D. Zheleznyakov[1] E. Kharlamov[1] V. Klungre[2] M. Skjæveland[2] D. Hovland[2]
M. Giese[2] I. Horrocks[1] A. Waaler[2]

[1] University of Oxford [2] University of Oslo

Abstract. In ontology-based data access (OBDA) the users access relational databases (RDBs) via ontologies that mediate between the users and the data. Ontologies are connected to data via declarative ontology-to-RDB mappings that relate each ontological term to an SQL query. In this demo we present our system KeywDB that facilitates construction of ontology-to-RDB mappings in an interactive fashion. In KeywDB users provide examples of entities for classes that require mappings and the system returnes a ranked list of such mappings. In doing so KeywDB relies on techniques for keyword query answering over RDBs. During the demo the attendees will try KeywDB with Northwind and NPD FP databases and collections of mappings that we prepare.

1 Introduction

Motivation. Ontology-based data access (OBDA) is a prominent approach to information integration in which an *ontology* that describes the domain of interest rather than the data is used to mediate between data consumers and relational data sources (RDBs). In OBDA data consumers are typically assumed to be domain experts who do not have a prior knowledge about the way the data is organised at the source. Thus, they access data by expressing their information needs as ontological queries. The ontology is connected to the data via a set of (*ontology-to-RDB*) *mappings*, declarative specification of the form $P(\vec{x}) \leftarrow \mathtt{sql}(\vec{x})$ that relate ontological terms P with SQL queries `sql` over the underlying data and that are used for automatic translation of ontological queries into data-level queries which can be executed by the underlying database management system.

Ontologies and mappings are clearly the main OBDA assets and thus acquiring them is of utter importance for deploying and maintaining any OBDA application. Ontologies capture domains of interest, they are data independant and thus they can be reused in different applications with the same domain. On contrary, mappings are hardly reusable since they depend on particular data sources. Therefore, in order to deploy an OBDA system over a given set of data sources, one has to develop a set of mappings specific for these sources. Building mappings manually is, however, a costly process, especially for large and complex databases (e.g., see [3, 4]).

In order to address this issue and facilitate mapping construction a number of approaches has been developed. Most of them focus on mappings of a specific form, called *direct mapping* [5], and under these approaches the mappings simply mirror the database schema by associating a table to a class and an attribute to a property. There are also approaches that allow to construct more complex mappings. For example, in [2], the system is able to compute *all* possible queries that involve joins between tables and equalities between column names and values (under certain restriction). The main problem of this kind of approaches is that the number of the returned mappings is huge and manually filtering in order to select the right mappings is an expensive procedure. So the existing approaches either compute a few simple mappings that are insufficient in many applications or too many complex mappings most of which are irrelevant for the application at the hand. Therefore, there is a need for techniques to facilitate mapping creation that are *precise* in the sense that they compute the mappings required in a concrete application.

Our Contribution. We propose a novel, semi-automatic approach for mapping construction that *(i)* allows for creation of mappings expressive enough to satisfy the users' information needs (that is more expressive that in the case of direct mappings) and *(ii)* does not overwhelm users with candidate mappings. We implemented our approach in the KeywDB system and will now explain the approach on the following scenario. Assume that the user wants to map a class to the data. KeywDB will help the user in the three following steps:

(i) Since the user is a domain expert, they know what objects the class should contain. Thus, KeywDB will ask the user to provide a *description* of several objects from the class, where a description is a set of keywords.

(ii) KeywDB will turn the input descriptions into a ranked list of queries and return the user top-k *candidate* queries, where k is fixed in advance.

(iii) The user will gives a *feedback* on the list by choosing those queries from the list that they think are correct.

In order to support this scenario we developed a formal semantics of transformation of descriptions into a ranked list of candidate queries, and introduced a query ranking model tailored towards our framework.

Demonstration Scenarios. We prepared two demonstration scenarios, which are based on the Northwind[1] and NPD FP [6] databases. A demo attendee will be able to create mappings for classes in each of the scenarios.

2 KeywDB System

Setting. Consider a scenario where a user is looking for a mapping for a class C to a relational database D. We assume that the user is a domain specialists and they know what kind of objects should be in C. Thus, they can describe several *examples* of such objects $o_1, \ldots, o_n$, each with a set K_i of *keywords* $\{k_1^i, \ldots, k_{n_i}^i\}$. To describe our approach we first need to define the following notions. Let S be a schema of D. A *schema graph* $G_s = (V_S, E_S)$ is a graph where V_S is set of relations of S and (R_i, R_j) is in E_S if and only if there is a primary to foreign

[1] https://northwinddatabase.codeplex.com/

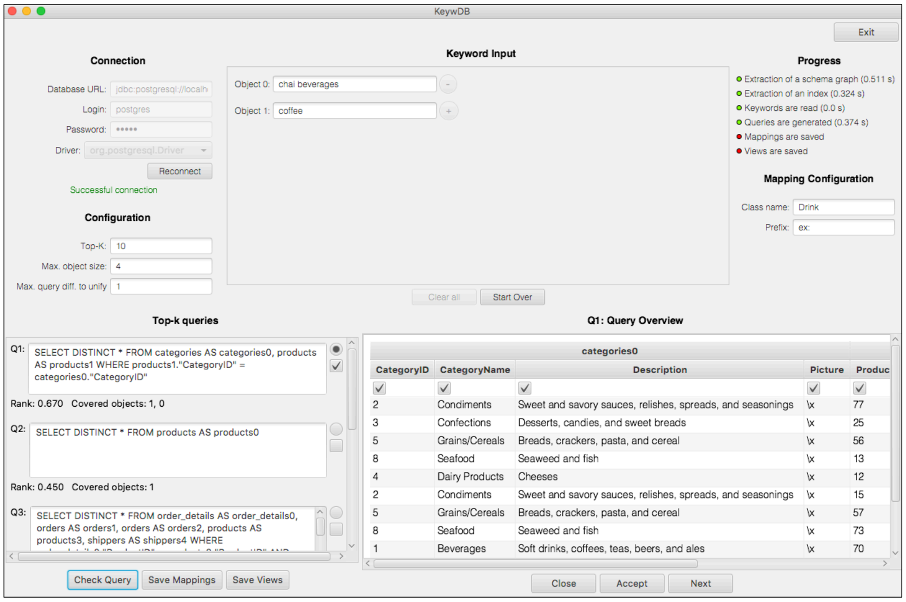

Fig. 1. A screenshot of the KeywDB system

key relationship between R_i and R_j. A *data graph* [1][2] G_D of the database D is a graph where V_D is a set of all tuples occurring in D and (t_i, t_j) is in E_D if and only if $t_i \in R_i$, $t_j \in R_j$ and $(R_i, R_j) \in E_S$.

Our Approach in a Nutshell. Having a set K_i of keywords describing an object o_i, we *extract* a ranked list of *candidate objects* from G_D, where each candidate object is a connected subgraph of G_D such that *(i)* every keyword from K_i is contained in at least one tuple of this subgraph[3], and *(ii)* it is minimal, that is, we cannot remove any tuple from it and still be connected and satisfying Condition (i). Then, each of the candidate objects o'_i is turned into a SQL query q'_i such that the answer $q'_i(D)$ over D contains o'_i, thus a ranked list of *candidate queries* is obtained. Note that *(i)* the rank of a candidate query q'_i is a function of the rank of the corresponding candidate object o'_i, and *(ii)* a candidate query may correspond to several candidate objects, in which case the rank of each of these objects influences on the rank of the query. Performing the same procedure for each K_i, we obtain a set of lists $L_1, \ldots, L_n$ of candidate queries. We unify them into a final list L, where the rank of each candidate query depends on *(i)* its rank in a list L_i it appears in and *(ii)* a number of such lists.

Ranking Model. In order to rank objects and then queries we rely on their several characteristics: on the size, diameter, and distribution of keywords over them.

[2] Note that in [1] a data graph is called a *joining network of tuples*.

[3] A tuple contains a keyword if the latter one appears in an attribute of the former one.

3 Demonstration Scenario

We prepared two databases on which our system can be tested. The first database, Northwind, contains the sales data for a fictitious company called Northwind Traders, which imports and exports speciality foods from around the world. The second one is Norwegian Petroleum Directorates FactPages (NPD FP) [6], a Norwegian public information repository about the oil and gas sector.

During the demo KeywDB will be available in two scenarios.

(S1) *Supervised:* We prepared 20 goal mappings for 20 classes for each database. For each class, the system will automatically generate keyword descriptions of one, two or three different objects that the class is supposed to contain. The attendee will be demonstrated whether the top-k mapping returned by the system contain the corresponding goal mapping, where $k = 1$, 3 and 5.

(S2) *Unsupervied:* The attendee will be able to explore the schema and create themselves a class they would like to build a mapping for. Additionally, for each database, 10 classes, not linked to the database, and their intuitive descriptions will be provided. Then, the user will be able to explore the data and compose descriptions of objects for both their and prepared class.

In Figure 1 there is a screenshot of KewDB where the user has been looking for a mapping for a class 'Drink'. The user provided examples of two objects: one is described with two keywords 'chai' and 'bevarage' and another with one keyword 'coffee'. KeywDB in turn returned several mappings, e.g., the mapping with the following query is returned first and has the rank equal to 0.670:

```
SELECT DISTINCT *
FROM categories AS categories0, products AS products1
WHERE products1."CatagoryID"=categories0."CategoryID"
```

4 References

[1] V. Hristidis and Y. Papakonstantinou. Discover: Keyword Search in Relational Databases. In: *VLDB*. 2002.

[2] E. Jiménez-Ruiz, E. Kharlamov, D. Zheleznyakov, I. Horrocks, C. Pinkel, M. G. Skjæveland, E. Thorstensen, and J. Mora. BootOX: Practical Mapping of RDBs to OWL 2. In: *ISWC*. 2015.

[3] E. Kharlamov, D. Hovland, E. Jiménez-Ruiz, D. Lanti, H. Lie, C. Pinkel, M. Rezk, M. G. Skjæveland, E. Thorstensen, G. Xiao, D. Zheleznyakov, and I. Horrocks. Ontology Based Access to Exploration Data at Statoil. In: *ISWC*. 2015.

[4] E. Kharlamov, N. Solomakhina, Ö. L. Özçep, D. Zheleznyakov, T. Hubauer, S. Lamparter, M. Roshchin, A. Soylu, and S. Watson. How Semantic Technologies Can Enhance Data Access at Siemens Energy. In: *ISWC*. 2014.

[5] J. Sequeda, S. H. Tirmizi, O. Corcho, and D. P. Miranker. Survey of Directly Mapping SQL Databases to the Semantic Web. In: *KER* 26.4 (2011).

[6] M. G. Skjæveland, E. H. Lian, and I. Horrocks. Publishing the Norwegian Petroleum Directorate's FactPages as Semantic Web Data. In: *ISWC*. 2013.