On demand translation for querying incompletely aligned datasets

Ana I. Torre-Bastida, Jesús Bermúdez, Arantza Illarramendi, Marta González

UPV/EHU / LSI / TR 02-2014

On demand translation for querying incompletely aligned datasets

Technical Report. November 2014
Departamento de Lenguajes y Sistemas Informáticos. UPV/EHU.

Ana I. Torre-Bastida, Jesús Bermúdez,
Arantza Illarramendi, Marta González

Abstract

More and more users aim at taking advantage of the existing Linked Open Data environment to formulate a query over a dataset and to then try to process the same query over different datasets, one after another, in order to obtain a broader set of answers. However, the heterogeneity of vocabularies used in the datasets on the one side, and the fact that the number of alignments among those datasets is scarce on the other, makes that querying task difficult for them.

Considering this scenario we present in this paper a proposal that allows on demand translations of queries formulated over an original dataset, into queries expressed using the vocabulary of a targeted dataset. Our approach relieves users from knowing the vocabulary used in the targeted datasets and even more it considers situations where alignments do not exist or they are not suitable for the formulated query. Therefore, in order to favour the possibility of getting answers, sometimes there is no guarantee of obtaining a semantically equivalent translation.

The core component of our proposal is a query rewriting model that considers a set of transformation rules devised from a pragmatic point of view. The feasibility of our scheme has been validated with queries defined in well known benchmarks and SPARQL endpoint logs, as the obtained results confirm.

1 Introduction

People are witnessing an explosion of types, availability and volume of data sources accessible in the Web. In particular the so called Web of Data can be considered one of the major global repositories in which the number of available linked datasets is continuously increasing, mainly promoted by initiatives such as Linked Open Data, Open Government and Linked Life Data. One main objective of these initiatives is to open up data silos and to publish their contents

in a semi structured format with links between related data entities. As a result a growing number of Linked Open Data sources (from diverse provenance and about different domains) are made available which can be freely browsed and searched to find and extract useful information.

In this new scenario, users and more particularly scientists, envision new opportunities to advance faster in their research accessing available sources. However, access to them is difficult for the users. Difficulties are mainly related to the highly distributed structure and evolving nature of the environment. Aspects related to volume (the number of datasets is large and it is difficult to know of their existence), dynamism (datasets evolve quickly and are added and removed over time) and heterogeneity (datasets vary in size, there is no standard for source descriptions, and access options vary) present unique research challenges. In this paper, we mainly focus on the heterogeneity aspect. It is quite common for several datasets to describe the same or overlapped domains (for example Linked GeoData[1] and Geo Linked Data[2] in the geographic domain) but then use different vocabularies to describe similar information.

The development of systems that allow an easy access to information coming from different data sources, distributed over the internet, has been and still is considered a relevant research topic in the specialized literature. For example, a large variety of strategies have been proposed for distributed and federated databases (e.g. [1, 2, 3]) and in recent times for distributed RDF data sources (e.g. [4, 5]). Although some research challenges are similar in both scenarios, there are also significant differences, as presented in [6].

In this paper we present a proposal that allows a user first, to formulate a query over a source dataset she/he is familiar with, and then, to enrich the obtained answer by accessing other different datasets, on demand of the user, without having to be aware of their internal structure. The novel contributions of our proposal are presented in the following paragraphs.

A friendly and incremental query answering process The user formulates queries in a selected source vocabulary, then selects the target datasets where the query must be evaluated. Our system is in charge of navigating through those target datasets, one by one, providing the answers in an incremental way. Our system tries to faithfully translate the formulated query but, sometimes, due to a mismatch of dataset vocabularies or due to an incomplete definition of alignment axioms, it is not possible to guarantee a translation that preserves the semantics of the original query. Then, our approach offers a non-semantics-preserving translation but an acceptable and effective enough one, that we think is better than not providing a translation at all, because that translation can be used by the user for getting more answers. In the case of not semantics preserving translations, our system provides to the user with a measure of the semantic similarity between the original query and the translation, but the computation and processing of this similarity factor is out of the scope

[1] http://linkedgeodata.org/
[2] http://geo.linkeddata.es/

of this paper.

On the fly management of a wide range of transformation rules Translation across datasets is achieved through the management of a wide range of transformation rules. Apart from the rules based on classical mappings defined between datasets (synonyms, hyponyms, hypernyms), the system also deals with EDOAL (Expressive and Declarative Ontology Alignment Language) [7] alignment rules and some other heuristic based rules which conforms a carefully controlled set of cases (answer-based rules, profile-based rules and feature-based rules). This allows a greater range of query transformations and therefore the chance of obtaining new answers increases. Rules are applied on the fly during the query processing task, taking into account the already existing mapping information that at that time is at hand.

Finally, we want to mention that the proposal has been validated with a prototype implementation that processes queries that appear in well known benchmarks such as QALD [3] and FedBench [8] and in SPARQL endpoints logs(DBpedia[4] and BNE[5]). The results of the validation process are promising and are presented in section 6.

In section 2 we present some works related with our proposal for querying distributed and heterogeneous datasets. Then, in section 3 we introduce some basic concepts and notation used throughout the paper. Next, in section 4 we explain the query rewriting model proposed. Later on, we show an overview of the query translation process at section 5 and experimental results are displayed in section 6. We finish with conclusions and future work.

2 Related work

The problem of query processing over linked data sources has been considered from different perspectives, such as the development of graphical user interfaces, to facilitate query formulation and architectural issues.

Among the works that try to develop a query tool, we can mention the following: 1) PowerAqua [9] which is an ontology based question answering system that offers a Natural Language query interface, which is able to locate and integrate information that can be massively distributed across heterogeneous data resources, and return answers. 2) AUTOSPARQL [10], SINA [11] whose goal is to convert keywords or natural language expressions to a SPARQL query. 3) SWIP system [12] that allows for querying RDF data from natural language-based queries. SWIP is based on the use of query patterns that characterize families of queries and that are instantiated with respect to the initial user query expressed in natural language (these patterns are specific to the vocabulary used to describe the data source to be queried. For rewriting query patterns, they

[3] http://nlp.uned.es/clef-qa/
[4] http://dbpedia.org
[5] http://datos.bne.es

experiment ontology matching approaches in order to find complex correspondences between different ontologies). 4) Query Med [13] that allows users to query multiple biomedical data sources providing keywords as input.

In our approach the starting point is a SPARQL query formulated over a source dataset. It can be argued that our proposal is less flexible at the time of query formulation. However, in contrast to many other works, our proposal does not impose restrictions concerning the types of queries that can be formulated and moreover, our query rewriting model considers a broader spectrum of query translations as we explain at section 4. Finally our approach could benefit from some advantages provided by the mentioned studies.

With respect to the underlying architecture two broad classes of approaches can be distinguished: *centralized repositories*, where several datasets are collected in advance, preprocessed and stored in centralized repositories and where the queries are evaluated against those centralized repositories, and *distributed* query processing, where queries are evaluated against the distributed sources. Within the last type of approach two more alternatives can be distinguished: *federated* query processing (e.g. [14, 4, 15]) in which a query against a federation of data sources is split into queries that can be answered by the individual data sources, and *explorative* query processing (e.g. [16])), where a query is first evaluated on an initial data source and then the Web of Data is explored by traversing interesting links pointing to other data sources which may contain more data entities satisfying the query. An interesting variant in the latter case are the so called *index-based* approaches (e.g. [17, 18]), that ignore the existence of links during the query execution process and rely on a pre-populated index which is used for identifying URIs to look up during query execution time. If we compare our approach with the mentioned approaches, we can say that it is more flexible in the sense that, on the one hand, it does not require a costly preprocessing task of data sources nor the management of synchronization techniques, as in the case of centralized repositories; and on the other hand, it does not require to perform the complicated task of building a federation. Moreover, as it happens in the case of the explorative query processing approach, using our proposal, the user formulates a query over a source dataset with which he is familiar and then it offers the possibility of enriching the obtained answer by accessing other different data sources (in an incremental way, one data source at a time) in a transparent way; that is, without being aware of the internal structure of the data sources. The main difference in this case is that our proposal is able to deal, during the navigation process, with datasets supported by heterogeneous vocabularies. It also manages a wide range of query transformation rules (not only equivalence type mappings in contrast to many of the existing approaches), although it does not deal with weighted ontology mappings as the work presented in [15] does.

Regarding semantics-preserving query translation, we can find works such as [5] that provide a generic method for SPARQL query rewriting with respect to a set of predefined mappings between ontology schemas, being the work in [19] the most similar to our spirit of running the same query over different datasets. In this last case, they are more concerned with the expressiveness and some

limitations of their selected alignment model. In particular, they do not tackle properly the FILTER clause of SPARQL queries. Our proposal deals with a scenario where a translation that preserves the semantics is not a requirement and therefore it obtains semantics-preserving and not-semantics-preserving translations in order to increase the opportunities of getting translations.

Some other works [20, 21, 22] consider the relaxation of constraints in the query. In [20], the authors promote query relaxation on the fly, that is, the provision of approximate answers by relaxing the query conditions during query execution. In[21], relaxation is controlled by conditions from domain and user preference ontology; and in [22], the authors propose a SKOS-based term expansion and scoring technique to improve retrieval effectiveness. However, all of these approaches consider only a fixed source dataset for the relaxed query. That is to say, no change of vocabulary is considered. In contrast, our system handles different vocabularies and eventually with incompletely aligned datasets.

Finally, to deal with the diversity of representations of identical data items across different data sources, our proposal takes advantage of services such as Balloon [23] and SameAs (interlinking the Web of Data) [24], that manage coreference relationships.

3 Preliminaries

In this section we briefly introduce key concepts and notation used throughout the rest of the paper. For a complete definition of RDF and SPARQL we refer to [25, 26].

Let I be the set of all *IRIs* (Internationalized Resource Identifiers), L be the set of RDF *literals*, and B be the set of RDF *blank nodes*. These three infinite sets are pairwise disjoint. An RDF *triple* is a tuple $(s, p, o) \in (I \cup B) \times I \times (I \cup B \cup L)$; s is called the *subject*, p is called the *predicate*, and o is called the *object* of the triple, respectively. A finite set of triples can be represented as a directed edge-labelled graph where subjects and objects are nodes and edges are labelled by predicates. An RDF *graph* is a finite set of triples. For the purpose of this paper, a *Dataset* is an RDF graph. Given a dataset D, we refer to the set *voc(D)* $\subseteq (I \cup L)$ of IRIs and literals occurring in D as the *vocabulary* of D. We use the words *term* or *resource* to refer to elements in $I \cup L$.

SPARQL is the standard query language for RDF. The core of a SPARQL query is a *graph pattern*, which is used to match an RDF graph in order to search for the required answers. Let V be an infinite set of *variables*, disjoint from $(I \cup B \cup L)$. Variables in V are denoted by prefixing them with a question mark (for example, $?x$). Within this paper, a *triple pattern* is a tuple in $(I \cup V) \times (I \cup V) \times (I \cup L \cup V)$. That is to say, a triple pattern is a triple, without blank nodes, where a variable may occur in any place of the triple. And a *Graph pattern* is an expression recursively defined as follows:

- a triple pattern is a graph pattern.

- if P_1, P_2 are graph patterns, then $(P_1$ AND $P_2)$, $(P_1$ UNION $P_2)$, and $(P_1$ OPT $P_2)$ are graph patterns.
- if P is a graph pattern and C is SPARQL constraint, then $(P$ FILTER $C)$ is a graph pattern.

Within the scope of this paper, a query is defined by a pair $Q = (D, P)$ where D is the dataset to be used during the pattern matching and P is the graph pattern of the query. To define the semantics of a SPARQL query we refer to [27]. Here, we briefly present an intuitive notion. Let *vars(P)* be the set of variables occurring in P. The result of the evaluation of the pattern P against the dataset D is a set of bindings of the variables in *vars(P)* to elements in *voc(D)* in such a way that the graph pattern P, with each variable substituted by its corresponding binding and preserving the semantics of the SPARQL operators AND, UNION, OPT, and FILTER, matches the graph D.

A binding function over dataset D is a partial function $\mu\colon V \to$ *voc(D)* from the set of variables to the vocabulary of D. Let us abuse of notation. Given a triple $t = (s, p, o)$, we write $\mu(t)$ referring to the triple $(\mu(s), \mu(p), \mu(o))$ where μ is the identity function for terms that are not a variable. Given a set of triple patterns S, we write $\mu(S)$ referring to the set $\{\mu(t) \mid t \in S\}$. Then, a set of triple patterns S matches a graph D if there is a binding function $\mu\colon V \to$ *voc(D)* with *vars(S)* $\subseteq dom(\mu)$ such that $\mu(S) \subseteq D$.

Given a query $Q = (D, P)$ we write *voc(Q)* to denote the *vocabulary of the query* and *voc(Q)* = *voc(D)*. We write *voc(P)* $\subseteq (I \cup L)$ to denote the set of IRIs and literals occurring in the pattern P.

Definition 3.1. A query $Q = (D, P)$ is *adequate* if *voc(P)* $\subseteq$ *voc(Q)*.

The aim of this paper is to present and to evaluate a process that takes a given adequate query $Q_s = (D_s, P_s)$ (suitable for a *source* dataset D_s), and translates it into another adequate query $Q_t = (D_t, P_t)$ in order to be suitably evaluated over a selected *target* dataset D_t. The translating process produces Q_t as a semantically equivalent query to Q_s as long as enough equivalence mappings between *voc(D_s)* and *voc(D_t)* are found. Those mappings may come from whatever accessible device: VoID[6] linksets, co-reference services, mapping services, etc. The distinguishing point is that the process produces a mimetic query Q_t even in the case when no equivalent translation for Q_s is found. That is to say, sometimes the translation is not semantics-preserving due to our goal of producing a query fully adequate for the target dataset demanded by the user. The process is based on the graph pattern rewriting rules that will be presented in the next section.

4 Query rewriting model

This section presents a set of rules devised in order to rewrite a query graph pattern in a stepwise way towards the goal of being properly evaluated within

[6] http://www.w3.org/TR/void/

a targeted dataset, different from the source dataset of the original query.

The rule system has been devised from a pragmatic point of view. The rules set up common sense heuristics to obtain acceptable translations even when no semantically equivalent translations are at hand (due to vocabulary mismatch, for instance). The rules can be easily implemented and can be efficiently processed as will be shown in section 6. Moreover, again for pragmatic reasons, preconditions for the application of the rules take into account a carefully restricted context of the terms occurring in the graph pattern. Although restricted, the system has shown to be quite effective achieving acceptable translations (see section 6). Nevertheless, the system can be easily extended with additional rules.

Before presenting the rule model we need to introduce some notation. Rewriting rules need to express triple transformation by substitution of any of its components.

Definition 4.1. Let (s, p, o) be a triple pattern. *Substitution* of term v by term x is defined by the following function

$$(s,p,o)[v/x] = \begin{cases} (x,p,o) & \text{if } v = s \\ (s,x,o) & \text{if } v = p \\ (s,p,x) & \text{if } v = o \\ (s,p,o) & \text{otherwise} \end{cases}$$

Rewriting rules will be applied to graph patterns.

Definition 4.2. Let P be a graph pattern. *Replacement* of triple pattern q by a graph pattern r is defined by the following function

$$P[q/r] = \begin{cases} P & \text{if } P \text{ is a triple pattern and } P \neq q \\ r & \text{if } P \text{ is a triple pattern and } P = q \\ P_1[q/r] \ op \ P_2[q/r] & \text{if } P = P_1 \ op \ P_2 \text{ with } op \in \{\text{AND, UNION, OPT}\} \\ P_1[q/r] \text{ FILTER } C[q/r] & \text{if } P = P_1 \text{ FILTER } C \end{cases}$$

The rules are composed of two clauses, the left hand side (LHS) and the right hand side (RHS). The LHS clause presents a pattern which is to be matched to a subgraph in the graph pattern that must be rewritten. LHS has two parts that we call pattern QP and context $\{QC\}$. QP is a graph pattern and $\{QC\}$ is a constraint predicate that relates terms occurring in QP and a set of triples from the close context of terms occurring in the query pattern P. The RHS clause is a surrogate graph pattern RP which will replace the subgraph matched by QP. We write the rule in the following form: $QP\{QC\} \rightarrow RP$. This means that, if in the graph pattern to be rewritten there is a subgraph matched by QP and the query constraint QC is satisfied, then the result of applying the rule is $P[QP/RP]$.

We consider five kinds of rules, each kind based on a different motif: equivalence, hierarchy, answer-based, profile-based, and feature-based. Moreover, each rule application carries a similarity factor, whose value depends on the elements involved in the replacement. That factor is kept associated to the rewritten

element of the original query pattern, and it is meant to reflect a similarity measure between the replaced term and the replacement after application of the rule. The explanation of the detailed computation of that factor is out of the scope of this paper. Let us just point out that it is a value in the closed real interval $[0, 1]$.

Now, let us consider a pragmatic scenario in which a bridge dataset is taken into account. In order to favour the possibilities of finding alignments we admit mappings between both the original and target dataset and a bridge dataset. That scenario is quite frequent, since in almost any domain there is a popular dataset that may play such a reference role.

In the following subsections each kind of rule is explained and motivated. Generic *v:*-prefixed letters *v:s, v:p, v:o* are used to refer to terms belonging to vocabulary $voc(D_v)$. Specifically, *s:*-prefix, *t:*-prefix, and *b:*-prefix refer to the source, target, and bridge vocabulary, respectively. No syntactic relationship is assumed among *s:u, t:u*, and *b:u*. Notice that the aim of the system is to translate a SPARQL query expressed in terms of a source dataset D_s into another query expressed in terms of a target dataset D_t, perhaps using a bridge dataset D_b.

Notice also that RDF literals occurring in the queries are not rewritten with the following rules, but with string transformation functions properly constructed. We consider that, previous to the process of rules application, every literal s occurring in the graph pattern of the original query is replaced by the corresponding literal $f_t(s)$ in the target dataset D_t. Therefore, in the context part of the rules, the reference for a non adequate term is always $u_i \in (I - voc(D_t))$.

4.1 Equivalence rules

Equivalence rules apply when non adequate terms (i.e. terms not belonging to the target dataset D_t) occurring in the current query are involved in individual equivalence mappings such as `owl:sameAs`, or structural equivalence mappings such as those captured by the EDOAL language. The aim of these rules is to transform a query into an equivalent one.

First of all, the system takes advantage of equivalence alignments obtained by any pattern matching system (for instance, LogMap[7], Scarlet [28], Blooms[8], or any else). Assume those alignments are captured as EDOAL alignments where the registered relation is `Equivalence`. Then, if <ELHS> and <t:ERHS> (i.e. every term in `ERHS` is in D_t) are, respectively, the left and right hand side of an EDOAL equivalence rule, the system incorporates the rule:

$$\texttt{ELHS} \rightarrow \texttt{t:ERHS} \tag{1}$$

Secondly, individual equivalence mappings between adequate and non adequate terms are taken into account by the following rule. The generic predicate *eq* represents a wildcard for any predicate of an extensible set of equivalence

[7] http://www.cs.ox.ac.uk/projects/LogMap/
[8] http://semanticweb.org/wiki/BLOOMS

predicates such as {`owl:sameAs, skos:exactMatch, owl:equivalentClass, owl:equivalentProperty`}. Notice that *eq* is symmetric, and therefore triple (*a,eq,b*) means the same as triple (*b,eq,a*).

$$(u_1, u_2, u_3)\{\exists i.\ u_i \in (I - voc(D_t)) \wedge \forall k \in \{1, \ldots, n\}.(u_i, eq, t\colon u_k)\} \\ \rightarrow \text{UNION}_{k=1\ldots n}(u_1, u_2, u_3)[u_i/t\colon u_k] \quad (2)$$

Thirdly, a selected bridge dataset D_b may be considered to help in finding more equivalence mappings. This is captured by the following rule:

$$(u_1, u_2, u_3)\{\exists i.\ u_i \in (I - voc(D_t)) \wedge \\ \forall k \in \{1, \ldots, n\}.(u_i, eq, b\colon u_k) \wedge (b\colon u_k, eq, t\colon u_k)\} \\ \rightarrow \text{UNION}_{k=1\ldots n}(u_1, u_2, u_3)[u_i/t\colon u_k] \quad (3)$$

Equivalence mappings may be taken from VoID linksets associated to any of the involved datasets or retrieved from any co-reference service system such as sameAs.org[9] or Balloon[10].

4.2 Hierarchy rules

These kind of rules transform the query by generalising or specialising non adequate terms for which equivalence rules did not succeed in their translation. The aim of these rules is to construct a looser or tighter query when the known mappings do not inform of direct equivalences.

We use a generic predicate *sub* to represent a wildcard for any predicate of an extensible set of hierarchy predicates such as {`skos:narrower, skos:broader, rdfs:subClassOf, rdfs:subPropertyOf`}. The system considers two basic possibilities: (1) A term in a triple pattern is known to be a subterm of a collection of adequate terms, then the triple pattern will be replaced by the conjunction (i.e. AND operator) of a looser triple patterns, and (2) A term in a triple pattern is known to be a superterm of a collection of adequate terms, then the triple pattern will be replaced by the disjunction (i.e. UNION operator) of a tighter triple patterns.

$$(u_1, u_2, u_3)\{\exists i.\ u_i \in (I - voc(D_t)) \wedge \\ \forall k \in \{1 \ldots n\}(u_i, sub, t\colon v_k) \vee \\ (\ (u_i, sub, v) \wedge (v, sub, t\colon v_k)\ ) \vee \\ (\ (u_i, sub, v) \wedge (v, eq, t\colon v_k)\ )\} \\ \rightarrow \text{AND}_{k=1\ldots n}(u_1, u_2, u_3)[u_i/t\colon v_k] \quad (4)$$

[9] http://sameas.org/

[10] http://schlegel.github.io/balloon/balloon-fusion.html

$$
\begin{aligned}
(u_1, u_2, u_3)\{\exists i.\ u_i \in (I - voc(D_t)) \wedge \\
\forall k \in \{1 \ldots n\}(t\colon v_k, sub, u_i) \vee \\
(\ (t\colon v_k, sub, v) \wedge (v, sub, u_i)\ ) \vee \\
(\ (t\colon v_k, eq, v) \wedge (v, sub, u_i)\ )\} \\
\rightarrow \text{UNION}_{k=1\ldots n}(u_1, u_2, u_3)[u_i/t\colon v_k]
\end{aligned} \tag{5}
$$

This kind of rules replaces a triple pattern by a looser or tighter pattern.

4.3 Answer-based rules

It is possible that after applying equivalence and hierarchy rules some non adequate terms remain in the query pattern. Instead of abandoning the translation, some heuristics are used to try to obtain mimetic translations. The following kind of rules use resources, that are answers to the query in the source dataset, as examples of what the query is looking for in the target dataset. Triples involving those resources in the target dataset are used to mimic the triple pattern to be replaced. The intuition is that triples stated about the answer samples in the target dataset probably resemble expected answers of the original query.

Let A be a set of resources. Let us define the set $\mathcal{ID}_t(A)$ of resources in target vocabulary that are known to be the same (modulo equivalence) as some resource in a given set A of resources.

$$\mathcal{ID}_t(A) = \{b \in voc(D_t) \mid \exists a \in A.(a, eq, b) \in \mathcal{E}(A, D_t)\}$$

Let $\mathcal{A}(Q_s, ?x)$ be the set of value bindings to variable $?x$ (i.e. the resource answer set for $?x$) when evaluating query $Q_s = (D_s, P_s)$. Then $\mathcal{ID}_t(\mathcal{A}(Q_s, ?x))$ is the set of resources in the target dataset that are equivalent to some resource in the answer set for $?x$ after evaluation of the query on the source dataset.

Triple patterns looking for subject (resp. object) with adequate predicate $t\colon p$ and non adequate object (resp. subject) will be replaced by the union of triple patterns composed by samples of objects (resp. subjects) related with answers by the predicate $t\colon p$. Let us formalise it with the following rules:

$$
\begin{aligned}
(?x, t\colon p, u)\{u \in (I - voc(D_t)) \wedge \\
\forall k \in \{1 \ldots n\} t\colon s_k \in \mathcal{ID}_t(\mathcal{A}(Q_s, ?x)) \wedge \\
\bigwedge_{j=1\ldots m_k} (t\colon s_k, t\colon p, t\colon o_{kj})\} \\
\rightarrow \text{UNION}_{k=1\ldots n}(\text{AND}_{j=1\ldots m_k}(?x, t\colon p, t\colon o_{kj}))
\end{aligned} \tag{6}
$$

$$
\begin{aligned}
(u, t\colon p, ?x)\{u \in (I - voc(D_t)) \wedge \\
\forall k \in \{1 \ldots n\}.t\colon o_k \in \mathcal{ID}_t(\mathcal{A}(Q_s, ?x)) \wedge \\
\bigwedge_{j=1\ldots m_k} (t\colon s_{kj}, t\colon p, t\colon o_k)\} \\
\rightarrow \text{UNION}_{k=1\ldots n}(\text{AND}_{j=1\ldots m_k}(t\colon s_{kj}, t\colon p, ?x))
\end{aligned} \tag{7}
$$

Triple patterns looking for subject (resp. object) with non adequate predicate p and adequate object $t\colon o$ (resp. subject) will be replaced by the conjunction of triple patterns composed with the shared predicates of the triples where answers are subjects. Let us define the set $\mathcal{FO}_t(A, r)$ (we call it *fixed object*) of predicates of triples in D_t that are shared by every resource in A and have object r:

$$\mathcal{FO}_t(A, r) = \{p \in voc(D_t) \mid \forall a \in A.(a, p, r) \in D_t\}$$

Respectively, let us define the set $\mathcal{FS}_t(A, o)$ (we call it *fixed subject*) of predicates of triples in D_t that are shared by every resource in A and have subject r.

$$\mathcal{FS}_t(A, r) = \{p \in voc(D_t) \mid \forall a \in A.(r, p, a) \in D_t\}$$

Then, the rewriting rules are the following:

$$\begin{aligned}(?x, p, t\colon o)\{p \in (I - voc(D_t)) \wedge \\ \forall k \in \{1 \dots n\} t\colon p_k \in \mathcal{FO}_t(\mathcal{ID}_t(\mathcal{A}(Q_s, ?x)), t\colon o)\} \\ \rightarrow \text{AND}_{k=1\dots n}(?x, t\colon p_k, t\colon o)\end{aligned} \tag{8}$$

$$\begin{aligned}(t\colon s, p, ?x)\{p \in (I - voc(D_t)) \wedge \\ \forall k \in \{1 \dots n\} t\colon p_k \in \mathcal{FS}_t(\mathcal{ID}_t(\mathcal{A}(Q_s, ?x)), t\colon s)\} \\ \rightarrow \text{AND}_{k=1\dots n}(t\colon s, t\colon p_k, ?x)\end{aligned} \tag{9}$$

Triple patterns looking for subject (resp. object) and predicate with non adequate object o (resp. subject s) will be replaced by a triple pattern with subject (resp. object) and predicate variables and an adequate object (resp. subject) determined by majority of occurrences in triples describing the answer set. Let us denote $\mathcal{MFO}_t(A)$ (we call it *most frequent object*) the resource in D_t that occurs more frequently as object in triples in D_t whose subjects are the resources in the answer set.

$$\mathcal{O}_t(A) = \{r \in voc(D_t) \mid \exists s \in A. \exists p.(s, p, r) \in D_t\}$$

$$\begin{aligned}\mathcal{N}_t(r) = &\text{ number of occurrences of resource } r \text{ in triples in } D_t \\ &\text{where the subject is a member of } \mathcal{ID}_t(\mathcal{A}(Q_s, ?x)) \\ \mathcal{MFO}_t(A) = t\colon o \;\; &\text{such that } \forall r \in \mathcal{O}_t(\mathcal{ID}_t(\mathcal{A}(Q_s, ?x))). \mathcal{N}_t(t\colon o) \geq \mathcal{N}_t(r)\end{aligned}$$

Analogously, we denote $\mathcal{MFS}_t(A)$ to the most frequent subject in D_t that occurs more frequently as subject in triples in D_t whose objects are the resources in the answer set.

$$\begin{aligned}(?x, ?p, o)\{o \in (I - voc(D_t)) \\ \rightarrow (?x, ?p, \mathcal{MFO}_t(\mathcal{ID}_t(\mathcal{A}(Q_s, ?x))))\end{aligned} \tag{10}$$

$$\begin{aligned}(s, ?p, ?x)\{s \in (I - voc(D_t)) \\ \rightarrow (\mathcal{MFS}_t(\mathcal{ID}_t(\mathcal{A}(Q_s, ?x))), ?p, ?x)\end{aligned} \tag{11}$$

4.4 Profile-based rules

This kind of rules consider the triples in the source dataset describing each non adequate resource in the query pattern. Let us call $\mathcal{P}_s(x)$ the *profile* of a resource x in a dataset D_s. It is the set of resources that are related to x by triples in D_s.

$$\mathcal{P}_s(x) = \{v \in voc(D_s) \mid (\exists p.(x,p,v) \in D_s \vee (v,p,x) \in D_s) \\ \vee (\exists a.(a,x,v) \in D_s \vee (v,x,a) \in D_s)\}$$

If a resource v, in the profile of a non adequate resource u, is equivalent to a resource $t\colon v$ in the target dataset, and there is a resource $t\colon u$ in the profile of $t\colon v$, sufficiently similar to u, then u will be replaced by $t\colon u$.

We denote $maxSim(a,b,h)$ the predicate that is satisfied if b is the resource with greatest similarity factor with respect to a and that factor is greater than h.

$$\begin{aligned}(u,p,o)\{u \in (I - voc(D_t)) \wedge \\ (\ (u,s\colon q,s\colon a) \vee (s\colon a,s\colon q,u)\ ) \wedge (s\colon a,eq,t\colon a) \wedge \\ (\ (t\colon a,t\colon q,t\colon b) \vee (t\colon b,t\colon q,t\colon a)\ ) \wedge maxSim(u,t\colon b,threshold) \\ \rightarrow (u,p,o)[u/t\colon b]\end{aligned} \tag{12}$$

$$\begin{aligned}(s,p,u)\{u \in (I - voc(D_t)) \wedge \\ (\ (u,s\colon q,s\colon a) \vee (s\colon a,s\colon q,u)\ ) \wedge (s\colon a,eq,t\colon a) \wedge \\ (\ (t\colon a,t\colon q,t\colon b) \vee (t\colon b,t\colon q,t\colon a)\ ) \wedge maxSim(u,t\colon b,threshold) \\ \rightarrow (s,p,u)[u/t\colon b]\end{aligned} \tag{13}$$

$$\begin{aligned}(s,p,o)\{p \in (I - voc(D_t)) \wedge \\ (\ (s\colon a,p,s\colon b) \vee (s\colon b,p,s\colon a)\ ) \wedge (s\colon a,eq,t\colon x) \wedge \\ (\ (t\colon x,t\colon q,t\colon y) \vee (t\colon y,t\colon q,t\colon x)\ ) \wedge maxSim(p,t\colon q,threshold) \\ \rightarrow (s,p,o)[p/t\colon q]\end{aligned} \tag{14}$$

4.5 Feature-based rules

This rule is the last option if non adequate terms remain in the query pattern after the above rules have already been applied. In this case, the intuitive motif is to replace the non adequate term by a new variable (therefore, generalizing the query) but constraining that variable with features of the replaced term (that is to say, triples where the term is the subject).

Finally, equivalence and hierarchy rules (in that order) are applied to the resulting feature-based transformed query graph. After that, any residual non adequate triple pattern is removed from the graph pattern.

$$(u_1, u_2, u_3)\{\exists i.u_i \in (I - voc(D_t)) \wedge \bigwedge_{k=1\ldots n} (u_i, s\colon p_k, s\colon o_k)\}$$
$$\rightarrow (u_1, u_2, u_3)[u_i/?v_i]\text{AND}_{k=1\ldots n}(u_i, s\colon p_k, s\colon o_k)[u_i/?v_i]$$
$?v_i$ a new variable not occurring in the graph pattern. (15)

4.6 Rules application order

The rule system initially tries to translate an original query into a semantically equivalent one. Therefore, equivalence rules are applied first. And, specifically, in the order of their numbers (1), (2), and (3).

Furthermore, while non adequate terms remain in the graph pattern the system sequentially applies the different kinds of rules. The second kind of rules to be applied are hierarchy rules, looking for looser or tighter adequate terms that could replace non adequate terms. Equivalence and hierarchy rules apply certain semantic relationships between terms. If that is not enough, the system turns to similarity relationships (with uncertain semantic transfer) mostly based on examples of answers and profile of the terms.

The third kind of rules to be applied is answer-based rules and the fourth kind is the profile-based rules. This latter kind is the most computationally expensive of the four and that is the reason why it is pushed to backward position. However, as it will be shown in section 6, that kind of rules solved a significant portion of the queries in the experiment.

Finally, if the precedent four kinds of rules do not succeed in obtaining an adequate query, the system applies the feature-based rules as an ultima ratio.

5 Overview of the query translation process

This section presents a motivating example that illustrates the steps followed in the query translation process. Imagine a film reporter who wants to know people who acted on a film entitled `Gravity`. Since the reporter is familiar with the vocabulary of LinkedMDB[11] (a linked open dataset about movies), the following SPARQL query is written and submitted to our system, specifying LinkedMDB as the source dataset.

```
PREFIX movie: <http://data.linkedmdb.org/resource/movie/>
PREFIX dc: <http://purl.org/dc/terms/>
SELECT ?actor
WHERE
{  ?film dc:title  'Gravity' .
   ?film movie:actor ?actor . }
```

[11] http://linkedmdb.org/

The system issues the query to the corresponding SPARQL endpoint but, unfortunately, no element is received as an answer[12]. Then, the reporter demands issuing that query to the DBpedia dataset. The first step for the system is to parse the query and generate its graph representation. During that process, the terms `dc:title` and `movie:actor` are discovered as non adequate for DBpedia. Moreover, DBpedia literal transforming function is applied to `Gravity` and it results $f_{DBpedia}(\texttt{Gravity}) = \texttt{Gravity@en}$. Then, the rule application process is launched.

Equivalence rules are the first to be applied. EDOAL rules matching the graph pattern are searched in the mapping repository associated to the pair LinkedMDB and DBpedia. Since no matching rule is found, the system tries to apply the rule number (2). The search for triples of the form $(u, eq, t\colon u)$ where u is `dc:title` or `movie:actor` is implemented by parameterized SPARQL queries over the corresponding mapping repositories, specific for rule number (2), and by asking co-reference services. In this case, specific SPARQL queries returned an empty set as an answer. However, the service sameAs.org reported 12 synonyms for `dc:title`, two of them (`foaf:name` and `rdfs:label`) adequate for DBpedia. No adequate synonym was reported for `movie:actor`. Applying rule number (2), the graph pattern is transformed into:

```
{ { ?film rdfs:label 'Gravity'@en .}
  UNION
  { ?film foaf:name  'Gravity'@en .}
  ?film movie:actor ?actor . }
```

The next rule to be tried is number (3), but the search for triples of the form $(\texttt{movie:actor}, eq, b\colon u)$ and $(b\colon u, eq, t\colon u)$, using Freebase[13] as bridge dataset, does not succeed. Then, it is the turn for hierarchy rules (4) and (5), but their preconditions for the term `movie:actor` are not satisfied either. Next, answer-based rules (6) to (11) should be considered, but they cannot apply because the pattern part in the LHS of all those rules $(?x, t\colon p, u)$, $(u, t\colon p, ?x)$, $(?x, p, t\colon o)$, $(t\colon s, p, ?x)$, $(?x, ?p, o)$, and $(s, ?p, ?x)$, does not match the triple pattern `(?film, movie:actor, ?actor)`.

Then, profile-based rules get into the play. Rules number (12) and (13) cannot apply because the constraint conditions in the context part of the rules ask for a non adequate IRI subject (for rule (12)) or IRI object (for rule (13)) while the triple pattern to be matched `(?film, movie:actor, ?actor)` presents variables in these places. However, precondition for rule number (14) is satisfied due to triples in LinkedMDB and DBpedia as the following (among others):

```
PREFIX db-o: <http://dbpedia.org/ontology/>
PREFIX db-r: <http://dbpedia.org/resource/>
(movie:film/62333, movie:actor, movie:actor/338)
(db-r:Alastair_Mackenzie, db-o:starring, db-r:The_Last_Great_Wilderness)
(movie:actor/338, owl:sameAs, db-r:Alastair_Mackenzie)
(movie:film/1894, movie:actor, movie:actor/40969)
(db-r:Killer's_Kiss, db-o:producer , db-r:Stanley_Kubrick)
```

[12]The query was issued on 24th/7/2014.
[13]https://www.freebase.com/

```
(movie:film/1894, owl:sameAs, db-r:Killer's_Kiss)
(movie:film/10849, movie:actor, movie:actor/29437)
(db-r:The_Indian_Runner, db-o:director , db-r:Sean_Penn)
(movie:film/10849, owl:sameAs, db-r:The_Indian_Runner)
```

The similarity function between the term `movie:actor` and each term of the properties set {`db-o:starring, db-o:producer, db-o:director`,...} is evaluated[14] and it turns out that results that the greatest similarity is achieved with the term `db-o:starring` and its value is also above the threshold parameter. Then, substitution `(?film, movie:actor, ?actor) [movie:actor/db-o:starring]` is applied and the query graph pattern becomes:

```
{ { ?film rdfs:label  'Gravity'@en .}
  UNION
  { ?film foaf:name  'Gravity'@en .}
  ?film db-o:starring ?actor . }
```

which represents an adequate query for DBpedia dataset. Therefore, feature-based rules do not get into the play and rule applications reach the end.

Finally, the obtained translation and the result of its evaluation are shown in table 1.

Translation	**Results**
SELECT ?actor WHERE { { ?film rdfs:label 'Gravity'@en . } UNION { ?film foaf:name 'Gravity'@en . } ?film db-o:starring ?actor . }	<http://dbpedia.org/resource/Eric_Schaeffer> <http://dbpedia.org/resource/Krysten_Ritter> <http://dbpedia.org/resource/Ivan_Sergei> <http://dbpedia.org/resource/Ving_Rhames> <http://dbpedia.org/resource/Rachel_Hunter> <http://dbpedia.org/resource/Robyn_Cohen> <http://dbpedia.org/resource/James_Martinez_(actor)> <http://dbpedia.org/resource/Seth_Numrich> <http://dbpedia.org/resource/Sandra_Bullock> <http://dbpedia.org/resource/George_Clooney>

Table 1: Adequate query for DBpedia and its results

6 Evaluation of the proposal

In this section we present some features concerning the validation process of our proposal. First of all, we show how the queries that took part in the tests were selected and then, we discuss the results.

6.1 Selection of the considered query set

When selecting the queries, our aim was to get a set that would contain a broad spectrum of SPARQL query types. For that reason, we looked at two aspects when making our selection: place of provenance of the queries and their syntactic structure. Concerning provenance we selected on the one hand, queries

[14]It takes into account searches in Wordnet synsets and some other computations.

that appeared in well known benchmarks (we selected 6 queries from QALD[15] and 7 from FedBench[8]), and on the other hand, queries that belonged to LOD SPARQL endpoints logs (we selected 3 queries from Dbpedia log and 2 from BNE log). Notice that those 18 queries were formulated over heterogeneous domains such as bibliographic, life science, etc.

Regarding the syntactic structure, we can mention that a variety of the SPARQL operators and joins of variables appear in the queries. Moreover, although SELECT type queries are the only type considered, such option should not be seen as a limitation, because the focus of our proposal is to translate query patterns and to show that translation process can also be analogously applied to other types of queries such as ASK, CONSTRUCT or DESCRIBE.

Tables 3, 4, 5, 6, 7, and 8 in A present the initial set of queries, grouped by domain, with its Gold standard and the translation obtained by our system.

6.2 A discussion about results

Taking into account that our approach generates adequate translations (notice that those translations do not always preserve the semantics of the initial query), we decided to analyse the nature of those generated translations and the answers that they provided. For that, first of all, we asked some experts (those having experience in querying the source and target datasets) to express the original queries that took part in the tests in terms of the target datasets (we call them *Gold standard* queries[16]). Then, we run the translation and the Gold standard queries over the corresponding target datasets. The answers obtained running the translation queries were called *Retrieved results*, and those obtained running the Gold standard queries were called *Relevant results*. Using those result sets, we calculated the well known information retrieval measures: Precision, Recall, and F-measure (see table 2).[17]

$$\text{Precision} = \frac{|\text{Relevant} \cap \text{Retrieved}|}{|\text{Retrieved}|} \qquad \text{Recall} = \frac{|\text{Relevant} \cap \text{Retrieved}|}{|\text{Relevant}|}$$

$$\text{F-measure} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

An analysis of the results revealed the following considerations: In 12 out of 18 queries that made up the considered set of queries (that is, the 66.6% of the queries) the translation queries provided the same results as the corresponding Gold standard queries. For those queries (Q1, Q3, Q4, Q6, Q7, Q8, Q9, Q10, Q13, Q14, Q15, and Q18) the F-measure value was 1. A deeper analysis of those 12 queries also revealed that in eight of them, the translation was equal to the Gold standard expression (Q3, Q6, Q8, Q9, Q10, Q13, Q15, and Q18). Even more, in four of them (Q8, Q9, Q10, and Q13) the semantics of the original query

[15] http://nlp.uned.es/clef-qa/

[16] See A.

[17] The experiments were conducted on 24th/7/2014.

Queries						
Domain:	Media-Domain					
Queries	Q1	Q2	Q3	Q4	Q5	Q6
|Relevant results|	12	5	9	20	358	39
|Retrieved results|	12	4103	9	20	496	39
|Retrieved ∩ Relevant|	12	5	9	20	358	39
Precision	1	0.001	1	1	0.72	1
Recall	1	1	1	1	1	1
F-measure	1	0.002	1	1	0.83	1
Domain:	LifeScience-Domain					
Queries	Q7	Q8	Q9	Q10	Q11	Q12
|Relevant results|	1	6	1	173	1	11
|Retrieved results|	1	6	1	173	2	1
|Retrieved ∩ Relevant|	1	6	1	173	1	1
Precision	1	1	1	1	0.5	1
Recall	1	1	1	1	1	0.09
F-measure	1	1	1	1	0.66	0.16
Domain:	Bibliographic-Domain					
Queries	Q13	Q14	Q15	Q16	Q17	Q18
|Relevant results|	37	1	1	301	682	26
|Retrieved results|	37	1	1	4	19827	26
|Retrieved ∩ Relevant|	37	1	1	4	682	26
Precision	1	1	1	1	0.034	1
Recall	1	1	1	0.013	1	1
F-measure	1	1	1	0.026	0.066	1

Table 2: Accuracy metrics for the original queries set

was preserved by the translation because the similarity factor was equal to 1, and in the other four (Q3, Q6, Q15, and Q18) the semantic was not preserved. In the case of queries (Q1, Q4, Q7, and Q14) the translation was different from the Gold standard expression but notice that they provided the same answers.

Regarding the remaining queries (Q2, Q5, Q11, Q12, Q16, and Q17) the translations did not provided the same results as the corresponding Gold standards: the F-measure values was less than 1 and the translation and Gold standard queries were obviously different. However, we noticed that in none of them precision and recall were lost for the same query. Queries (Q12 and Q16) lost recall but preserved precision and queries (Q2, Q5, Q11, and Q17) lost precision but preserved recall.

Concerning running time, we distinguish between the time that our prototype implementation needs to generate the translations and the time to run them in the corresponding SPARQL endpoints. Figure 1 displays both times for each query of the experiment. Focussing only in the query rewriting time which is the time dedicated by our implementation we can observe at figure 2 that context-based rules are those that needed highest rewriting times, as for those rules, both datasets have to be consulted and the number of pairs of elements that need to be compared for measuring the similarity was usually high. On the opposite side are equivalence rules because the verification of the rules precondition is very fast in this case.

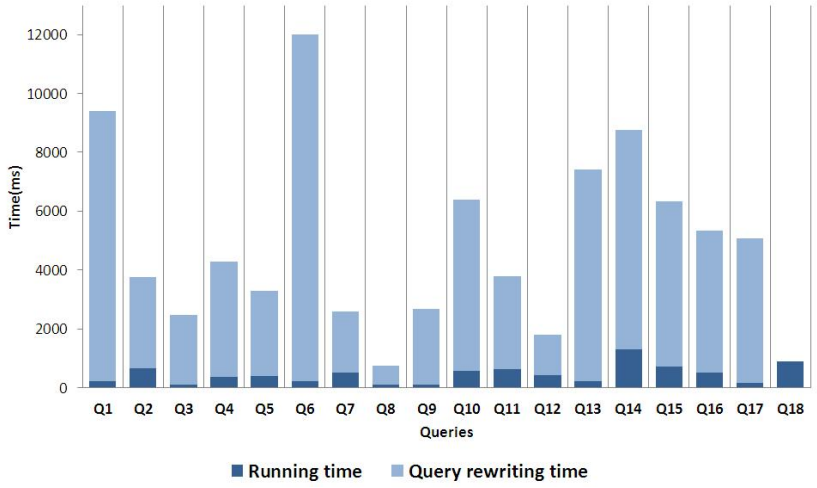

Figure 1: Query rewriting time vs. Running time at the SPARQL endpoint.

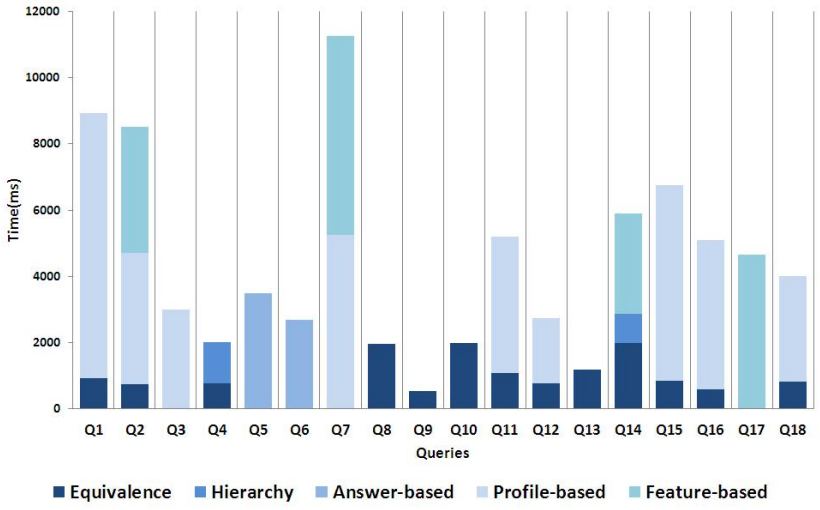

Figure 2: Rewriting time grouped by kinds of rules.

7 Conclusion and future work

Since the use of Linked Open datasets to extract actionable information is becoming very common for more and more people, the development of systems that provide an easy access and navigation among them is acquiring special relevance. In this paper we have presented a model that allows users to querying distributed and heterogeneous datasets abstracting from some technical issues such as the vocabulary heterogeneity. This way the users can use the vocabulary with which they are familiar to formulate the queries and the system is in charge of translating those query expressions according to the vocabulary of the target dataset. Moreover, the answer is upgraded in an incremental way.

During the translation process, a set of rewriting rules is used in our proposal. In addition to equivalence, hierarchical and EDOAL alignments among datasets (considered by a majority of distributed query processing approaches), our rules manage other types of information that enhance the possibilities of getting translations. Therefore, a novel contribution of the proposal presented in the paper is the definition and implementation of three new kinds of rules: answer-based rules, profile-based rules and feature-based rules respectively. These rules manage three different types of information as an example of what the query is looking for in the target dataset. In this way, answer-based rules use answers to the query in the source dataset; profile-based rules use triples in the source dataset describing each non-translated resource; and finally, feature-based rules generalize non-translated resources by variables, however constraining those variables in such a way that only triples where non-translated resources appear as subject are considered. The idea behind these rules is to capture semantics, that is relevant, and which is not explicitly expressed in the existing alignments.

Furthermore, in order to check the feasibility of our proposal we validated it not using synthetic examples but using queries extracted from well- known benchmarks and SPARQL endpoints logs. The validation process showed that about two thirds of the translations obtained equivalent results to those obtained by the Gold standard queries used for comparison, a fact that we consider encouraging. Moreover, the average running time for the query rewriting process is 5 sec., which can be considered acceptable in a real environment. As future work we plan to improve the current implementation in order to decrease the query rewriting time.

8 Acknowledgements

This work is supported by the TIN2013-46238-C4-1-R project, by the finantial grant IT797-13 and the Iñaki Goenaga (FCT-IG) Technology Centres Foundation.

References

[1] D. Kossmann, The state of the art in distributed query processing, ACM Computing Surveys (CSUR) 32 (4) (2000) 422–469.

[2] A. P. Sheth, J. A. Larson, Federated database systems for managing distributed, heterogeneous, and autonomous databases, ACM Computing Surveys (CSUR) 22 (3) (1990) 183–236.

[3] E. Mena, A. Illarramendi, V. Kashyap, A. Sheth, OBSERVER: An approach for query processing in global information systems based on interoperation across pre-existing ontologies, International journal on Distributed And Parallel Databases (DAPD) 8 (2) (2000) 223–272.

[4] B. Quilitz, U. Leser, Querying distributed rdf data sources with sparql, in: Proceedings of the 5th European semantic web conference on The semantic web: research and applications, Springer-Verlag, 2008, pp. 524–538.

[5] K. Makris, N. Bikakis, N. Gioldasis, S. Christodoulakis, Sparql-rw: transparent query access over mapped rdf data sources, in: Proceedings of the 15th International Conference on Extending Database Technology, ACM, 2012, pp. 610–613.

[6] O. Görlitz, S. Staab, Federated data management and query optimization for linked open data, in: New Directions in Web Data Management I, Vol. 1, Springer Verlag, 2011, pp. 109–137.

[7] J. David, J. Euzenat, F. Scharffe, C. Trojahn dos Santos, The alignment api 4.0, Semantic Web – Interoperability, Usability, Applicability 2 (1) (2011) 3–10.
URL `http://www.semantic-web-journal.net/sites/default/files/swj60_1.pdf`

[8] M. Schmidt, O. Görlitz, P. Haase, G. Ladwig, A. Schwarte, T. Tran, Fedbench: a benchmark suite for federated semantic data query processing, in: Proceedings of the 10th international conference on The semantic web-Volume Part I, 2011, pp. 585–600.

[9] V. Lopez, M. Fernández, E. Motta, N. Stieler, Poweraqua: Supporting users in querying and exploring the semantic web, Semantic Web 3 (3) (2012) 249–265.

[10] J. Lehmann, L. Bühmann, Autosparql: let users query your knowledge base, in: Proceedings of the 8th extended semantic web conference on The semantic web: research and applications-Volume Part I, 2011, pp. 63–79.

[11] S. Shekarpour, E. Marx, A.-C. N. Ngomo, S. Auer, Sina: Semantic interpretation of user queries for question answering on interlinked data, Web Semantics: Science, Services and Agents on the World Wide Web (0)

(2014) –. doi:http://dx.doi.org/10.1016/j.websem.2014.06.002.
URL http://www.sciencedirect.com/science/article/pii/S1570826814000468

[12] P. Gillet, C. T. dos Santos, O. Haemmerlé, C. Pradel, Complex correspondences for query patterns rewriting., Ontology Matching (2013) 49–60.

[13] O. Seneviratne, R. Sealfon, Querymed: An intuitive federated sparql query builder for biomedical rdf data.
URL http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.389.4282

[14] A. Schwarte, P. Haase, K. Hose, R. Schenkel, M. Schmidt, Fedx: optimization techniques for federated query processing on linked data, in: Proceedings of the 10th international conference on The semantic web-Volume Part I, 2011, pp. 601–616.

[15] T. Fujino, N. Fukuta, Utilizing weighted ontology mappings on federated sparql querying, in: W. Kim, Y. Ding, H.-G. Kim (Eds.), Semantic Technology, Lecture Notes in Computer Science, Springer International Publishing, 2014, pp. 331–347. doi:10.1007/978-3-319-06826-8_25.
URL http://dx.doi.org/10.1007/978-3-319-06826-8_25

[16] O. Hartig, Zero-knowledge query planning for an iterator implementation of link traversal based query execution, in: Proceedings of the 8th extended semantic web conference on The semantic web: research and applications-Volume Part I, Springer-Verlag, 2011, pp. 154–169.

[17] G. Ladwig, T. Tran, Linked data query processing strategies, in: Proceedings of the 9th international semantic web conference on The semantic web-Volume Part I, Springer-Verlag, 2010, pp. 453–469.

[18] Y. Tian, J. Umbrich, Y. Yu, Enhancing source selection for live queries over linked data via query log mining, in: Proceedings of the 2011 joint international conference on The Semantic Web, Springer-Verlag, 2011, pp. 176–191.

[19] G. Correndo, M. Salvadores, I. Millard, H. Glaser, N. Shadbolt, Sparql query rewriting for implementing data integration over linked data, in: Proceedings of the 2010 EDBT/ICDT Workshops, EDBT '10, ACM, New York, NY, USA, 2010, pp. 4:1–4:11. doi:10.1145/1754239.1754244.
URL http://doi.acm.org/10.1145/1754239.1754244

[20] K. B. Reddy, P. S. Kumar, Efficient trust-based approximate sparql querying of the web of linked data, in: Uncertainty Reasoning for the Semantic Web II, Springer, 2013, pp. 315–330.

[21] P. Dolog, H. Stuckenschmidt, H. Wache, Robust query processing for personalized information access on the semantic web, in: Proceedings of

the 7th international conference on Flexible Query Answering Systems, Springer-Verlag, 2006, pp. 343–355.

[22] B. Haslhofer, F. Martins, J. Magalhães, Using skos vocabularies for improving web search, in: Proceedings of the 22nd international conference on World Wide Web companion, International World Wide Web Conferences Steering Committee, 2013, pp. 1253–1258.

[23] K. Schlegel, F. Stegmaier, S. Bayerl, M. Granitzer, H. Kosch, Balloon fusion: Sparql rewriting based on unified co-reference information, in: Data Engineering Workshops (ICDEW), 2014 IEEE 30th International Conference on, IEEE, 2014, pp. 254–259.

[24] H. Glaser, A. Jaffri, I. Millard, Managing co-reference on the semantic web, in: WWW2009 Workshop: Linked Data on the Web (LDOW2009), 2009, event Dates: 20 April 2009.
URL http://eprints.soton.ac.uk/267587/

[25] R. Cyganiak, D. Wood, M. Lanthaler, RDF 1.1 Concepts and Abstract Sysntax, w3C Recommendation 25 February 2014 (2014).
URL http://www.w3.org/TR/2014/REC-rdf11-concepts-20140225/

[26] The W3C SPARQL Working Group. SPARQL 1.1 Overview, w3C Recommendation 21 March 2013.
URL http://www.w3.org/TR/sparql11-overview/

[27] J. Pérez, M. Arenas, C. Gutierrez, Semantics and complexity of sparql, ACM Trans. Database Syst. 34 (3) (2009) 16:1–16:45. doi:10.1145/1567274.1567278.
URL http://doi.acm.org/10.1145/1567274.1567278

[28] M. Sabou, M. d'Aquin, E. Motta, Scarlet: semantic relation discovery by harvesting online ontologies, in: Proceedings of the 5th European semantic web conference on The semantic web: research and applications, Springer-Verlag, 2008, pp. 854–858.

A Experimental queries

Query 1(FedBench CD4) - LinkedMDB	
SELECT ?actor WHERE ?film purlc:title 'Top Gun' . ?film movie:actor ?actor .	
Gold Standard - DBpedia: SELECT ?actor WHERE ?film foaf:name 'Top Gun' . ?film db-o:starring ?actor .	**Translation** - DBpedia: SELECT ?actor WHERE { ?film foaf:name 'Top Gun' .} UNION { ?film rdfs:label 'Top Gun' .} ?film db-o:starring ?actor .
Query 2(FedBench CD5) - LinkedMDB	
SELECT ?film ?director ?genre WHERE ?film movie:director ?director. ?film movie:country movie:country/IT. ?film movie:genre ?genre .	
Gold Standard - DBpedia SELECT ?film ?director ?genre WHERE ?film db-o:director ?director. ?film db-p:country 'Italy'. ?film db-o:genre ?genre .	**Translation** - DBpedia SELECT ?film ?director ?genre WHERE ?film db-o:director ?director. ?film db-p:country db-r:Italy. ?film ?p ?genre .
Query 3(QALD4) - MusicBrainz	
SELECT ?name ?band WHERE ?artist rdfs:label ?name. ?artist mo:member_of ?band. ?band rdfs:label 'The Beatles' .	
Gold Standard - DBpedia SELECT ?name ?band WHERE ?artist foaf:name ?name. ?band db-o:formerBandMember ?artist. ?band foaf:name 'The Beatles'.	**Translation** - DBpedia SELECT ?name ?band WHERE ?artist foaf:name ?name. ?band db-o:formerBandMember ?artist. ?band foaf:name 'The Beatles'.
Query 4(QALD4) - MusicBrainz	
SELECT ?album WHERE ?album rdf:type mo:Record . ?album foaf:maker ?artist . ?artist foaf:name Slayer .	
Gold Standard - DBpedia SELECT ?album WHERE ?album rdf:type db-o:Album . ?album db-o:artist ?artist. artist foaf:name 'Slayer'.	**Translation** - DBpedia SELECT ?album WHERE { ?album rdf:type db-o:Album . } union{ ?album rdf:type db-o:Single.} ?album db-o:artist ?artist. artist foaf:name 'Slayer'.
Query 5(DBPEDIA log) - DBpedia	
SELECT DISTINCT ?a WHERE ?a a db-o:Artist. ?a foaf:name ?n. FILTER (regex(?n, 'John','i')) .	
Gold Standard - LinkedMDB SELECT DISTINCT ?a WHERE ?a a foaf:Person . ?a rdfs:label ?n . FILTER (regex(?n, 'John','i')) .	**Translation** - LinkedMDB SELECT DISTINCT ?a WHERE { ?a a movie:editor .} UNION { ?a a db:movie/producer . } UNION { ?a a foaf:Person . } ?a rdfs:label ?n . FILTER (regex(?n, 'John','i')) .

Table 3: Query set - Media Domain 1

Query 6(DBPEDIA log) - DBpedia	
SELECT DISTINCT ?a WHERE db-r:The_Other_Side_of_the_Wind ?p ?a .	
Gold Standard - LinkedMDB SELECT DISTINCT ?a WHERE movie:46921 ?p ?a .	**Translation** - LinkedMDB SELECT DISTINCT ?a WHERE movie:46921 ?p ?a .

Table 4: Query set - Media Domain 2

Query 7(FedBench LS1) - Drugbank	
SELECT ?drug ?melt WHERE ?drug rdf:type drugbank:drugs . ?drug drugbank:meltingPoint ?melt .	
Gold Standard - DBPedia SELECT ?drug ?melt WHERE ?drug rdf:type db-o:drug . ?drug db-o:meltingPoint ?melt.	**Translation** - DBPedia SELECT ?drug ?melt WHERE ?drug rdf:type db-o:drug . ?drug ?p ?melt.
Query 8(FedBench LS2) - Drugbank	
SELECT ?predicate ?object WHERE drugbank-drugs:DB00201 ?predicate ?object .	
Gold Standard - KEGG SELECT ?predicate ?object WHERE kegg:D005281 ?predicate ?object .	**Translation** - KEGG SELECT ?predicate ?object WHERE kegg:D005281 ?predicate ?object .
Query 9(QALD4) - Drugbank	
SELECT ?x WHERE drugbank-drugs:DB00404 rdfs:label ?x .	
Gold Standard - SIDER SELECT ?x WHERE sider:2118 rdfs:label ?x .	**Translation** - SIDER SELECT ?x WHERE sider:2118 rdfs:label ?x .
Query 10(QALD4) - SIDER	
SELECT ?p1 ?y1 ?p2 ?y2 WHERE sider:1690 ?p1 ?y1 . sider:119607 ?p2 ?y2 .	
Gold Standard - Drugbank SELECT ?p1 ?y1 ?p2 ?y2 WHERE drugbank-drugs::DB00445 ?p1 ?y1 . drugbank-drugs::DB00580 ?p2 ?y2 .	**Translation** - Drugbank SELECT ?p1 ?y1 ?p2 ?y2 WHERE drugbank-drugs::DB00445 ?p1 ?y1 . drugbank-drugs::DB00580 ?p2 ?y2 .

Table 5: Query set - Life Science 1

Query 11(QALD4) - Drugbank	
SELECT DISTINCT ?v0 ?v1 WHERE { drugbank-drugs:DB00194 drugbank:molecularWeightAverage ?v0. OPTIONAL { drugbank-drugs:DB00194 drugbank: molecularWeightMono ?v1. } }	
Gold Standard - DBPedia SELECT DISTINCT ?v0 WHERE dbpedia-r:Vidarabine db-o:molecularWeight ?v0 .	**Translation** - DBPedia SELECT DISTINCT ?v0 ?v1 WHERE db-r:Vidarabine db-o:molecularWeight ?v0 . OPTIONAL {db-r:Vidarabine db-o:molecularWeight ?v1. } .
Query 12(QALD4) - SIDER	
SELECT DISTINCT ?x WHERE sider:8378 sider:drugName ?x	
Gold Standard - DBPedia SELECT DISTINCT ?x WHERE db-r:Allopurinol rdfs:label ?x.	**Translation** - DBPedia SELECT DISTINCT ?x WHERE db-r:Allopurinol db-o:tradename ?x.

Table 6: Query set - Life Science 2

Query 13(BNE log) - BNE	
SELECT DISTINCT ?p ?o WHERE bne-resource:XX1718747 ?p ?o .	
Gold Standard - BNB SELECT DISTINCT ?p ?o WHERE bnb:CervantesSaavedraMiguelde1547-1616 ?p ?o .	**Translation** - BNB SELECT DISTINCT ?p ?o WHERE bnb:CervantesSaavedraMiguelde1547-1616 ?p ?o .
Query 14(DBpedia log) - DBpedia	
SELECT DISTINCT ?label ?author WHERE db-r:Pride_and_Prejudice a db-o:Book . db-r:Pride_and_Prejudice foaf:name ?label . db-r:Pride_and_Prejudice db-o:author ?author . ?author a db:owl:Artist .	
Gold Standard - Cambridge SELECT DISTINCT ?label ?author WHERE cam:cambrdgedb_...79f80074f a owl:Thing . cam:cambrdgedb_...79f80074f rdfs:label ?label . cam:cambrdgedb_...79f80074f dct:creator ?author . ?author a foaf:Person .	**Translation** - Cambridge SELECT DISTINCT ?label ?author WHERE cam:cambrdgedb_...79f80074f a ?o . cam:cambrdgedb_...79f80074f rdfs:label ?label . cam:cambrdgedb_...79f80074f dct:creator ?author . ?author a foaf:Person.

Table 7: Query set -Bibliographic 1

Query 15(BNE log) - DBpedia	
SELECT ?a ?birth WHERE ?a db-o:writer db-r:Leo_Tolstoy . OPTIONAL { db-r:Leo_Tolstoy db-owl:dateOfBirth ?birth. }	
Gold Standard - BNE SELECT ?a ?birth WHERE bne-r:XX933715 fr:P2010 ?a . OPTIONAL { bne-r:XX933715 fr:P3040 ?birth. }.	**Translation** - BNE SELECT ?a ?birth WHERE bne-r:XX933715 fr:P2010 ?a . OPTIONAL { bne-r:XX933715 fr:P3040 ?birth. }.
Query 16(Fedbench) - DBLP	
SELECT * WHERE ?a akt:has-author dblp-r:person/100007. OPTIONAL { ?e akt:edited-by dblp-r:person/100007. }	
Gold Standard - DBpedia SELECT * WHERE ?a db-o:author db-r:Tim_Berners-Lee . OPTIONAL {?e ?p db-r:Tim_Berners-Lee .}	**Translation** - DBpedia SELECT * WHERE ?a db-o:author db-r:Tim_Berners-Lee . OPTIONAL {?e db-o:owner db-r:Tim_Berners-Lee .}
Query 17(Fedbench) - DBpedia	
SELECT DISTINCT ?a ?r WHERE ?a db-o:publishedIn ?r	
Gold Standard - DBLP SELECT DISTINCT ?a ?r WHERE ?a dblp:article-of-journal ?r	**Translation** - DBLP SELECT DISTINCT ?a ?r WHERE ?a ?p ?c
Query 18(Fedbench) - DBLP	
SELECT ?article WHERE ?article dblp:Article_IsWrittenBy ?author . ?author akt:full-name ?Author . FILTER (regex (?Author, 'Tanenbaum', 'i')) ORDER BY ?Author	
Gold Standard - BNB SELECT ?article WHERE ?author bnb:hascreated ?article . ?author foaf:name ?Author . FILTER (regex(?Author, 'Tanenbaum', 'i')) ORDER BY ?Author	**Translation** - BNB SELECT ?article WHERE ?author bnb:hascreated ?article . ?author foaf:name ?Author . FILTER (regex(?Author, 'Tanenbaum', 'i')) ORDER BY ?Author

Table 8: Query set - Bibliographic 2