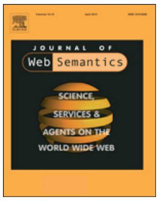
Web Semantics: Science, Services and Agents on the World Wide Web 37–38 (2016) 55–74

Contents lists available at ScienceDirect

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

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



Faceted search over RDF-based knowledge graphs[☆]

Marcelo Arenas[a], Bernardo Cuenca Grau[b], Evgeny Kharlamov[b], Šarūnas Marciuška[b], Dmitriy Zheleznyakov[b,*]

[a] Pontificia Universidad Catolica de Chile, Vicuna Mackenna 4860, Edificio San Agustin, Macul 7820436 Santiago, Chile

[b] University of Oxford, Department of Computer Science, Information Systems Group, Wolfson Building, Parks Road, Oxford OX1 3QD, UK

ARTICLE INFO

Article history:
Received 22 April 2015
Received in revised form
3 September 2015
Accepted 22 December 2015
Available online 31 December 2015

Keywords:
Faceted search
Ontology
OWL 2
RDF
SPARQL
Algorithms

ABSTRACT

Knowledge graphs such as Yago and Freebase have become a powerful asset for enhancing search, and are being intensively used in both academia and industry. Many existing knowledge graphs are either available as Linked Open Data, or they can be exported as RDF datasets enhanced with background knowledge in the form of an OWL 2 ontology. Faceted search is the de facto approach for exploratory search in many online applications, and has been recently proposed as a suitable paradigm for querying RDF repositories. In this paper, we provide rigorous theoretical underpinnings for faceted search in the context of RDF-based knowledge graphs enhanced with OWL 2 ontologies. We identify well-defined fragments of SPARQL that can be naturally captured using faceted search as a query paradigm, and establish the computational complexity of answering such queries. We also study the problem of updating faceted interfaces, which is critical for guiding users in the formulation of meaningful queries during exploratory search. We have implemented our approach in a fully-fledged faceted search system, SemFacet, which we have evaluated over the Yago knowledge graph.

1. Introduction

Knowledge graphs are large collections of interconnected entities enriched with semantic annotations, which have become powerful assets for enhancing search and are now widely used in both academia and industry. Prominent examples of large-scale knowledge graphs include Yago [1], Freebase [2], Google's Knowledge Graph [3], Facebook's Graph Search [4], Microsoft's Satori [5], and Yahoo's Knowledge Graph [6]. Many existing knowledge graphs are either available as Linked Open Data, or they can be exported as RDF datasets [7] enhanced with OWL 2 ontologies [8] capturing the relevant domain background knowledge.

SPARQL [9] has become the standard language for querying RDF data and OWL ontologies, and an increasing number of applications are relying on RDF, OWL 2, and SPARQL for storing, publishing, and querying data; in particular, access to knowledge graphs is often provided by a SPARQL endpoint. Writing SPARQL queries, however, requires some proficiency in the query language and is not well-suited for the majority of users [10,11]. Thus, an important challenge that has attracted a great deal of attention in the Semantic Web community is the development of simple yet powerful query interfaces for non-expert users [12–17]. This challenge becomes even more critical in the context of knowledge graphs such as Yago or Freebase, which are typically oriented towards end-users search.

Faceted search is a prominent approach for querying collections of entities where users can narrow down the search results by progressively applying filters, called *facets* [18]. A facet typically consists of a predicate (e.g., 'gender' or 'occupation' when querying entities about people) and a set of possible string values (e.g., 'female' or 'research'), and entities in the collection are annotated with predicate-value pairs. During faceted search users iteratively select facet values and the entities annotated according to the selection are returned as the search result.

Faceted search in the context of RDF has received significant attention and a number of systems have been developed [19–27]. Furthermore, several such systems have been successfully exploited for performing exploratory search over large knowledge graphs such as Freebase [28].

The theoretical underpinnings of faceted search in the context of RDF and knowledge graphs, however, remain relatively

[☆] This research was supported by the Royal Society, the EPSRC projects Score!, DBOnto, and MaSI^3 and the EU FP7 project Optique (n. 318338).

[*] Corresponding author.

E-mail addresses: marenas@ing.puc.cl (M. Arenas), bernardo.cuenca.grau@cs.ox.ac.uk (B. Cuenca Grau), evgeny.kharlamov@cs.ox.ac.uk (E. Kharlamov), sarunas.marciuska@cs.ox.ac.uk (Š. Marciuška), dmitriy.zheleznyakov@cs.ox.ac.uk (D. Zheleznyakov).

http://dx.doi.org/10.1016/j.websem.2015.12.002
1570-8268/

unexplored [10,29,30]. In particular, the following key questions have not been satisfactorily addressed in the literature (see our Related Work section):

(Q1) What fragments of SPARQL can be naturally captured using faceted search as a query paradigm?
(Q2) What is the complexity of answering such queries?
(Q3) What does it mean to generate and interactively update an interface according to a given RDF graph?

Questions 1 and 2 correspond to the study of the expressive power and complexity of query languages. These are central topics in data management, and addressing them is a key requirement to develop information systems that can provide correctness, robustness, scalability, and extensibility guarantees. Moreover, update (Question 3) is a key task in information systems where query formulation is fundamentally interactive. Our first goal is to answer these questions, thus providing rigorous and solid foundations for faceted search over RDF data.

Our second aim is to provide a framework for faceted search that is also applicable to the wider setting of OWL 2 and hence to ontology-enriched knowledge graphs such as Freebase and Yago. Existing works have focused mostly on RDF, thus essentially disregarding the role of OWL 2 ontologies. We see this as an important limitation. Ontological axioms not only can be used to enrich query answers over RDF datasets with implicit information, but also to enhance the navigation process by providing rich schema-level structure. Furthermore, RDF-based faceted search systems are data-centric and hence cannot be exploited to browse large ontologies such as SNOMED CT [31] or to formulate meaningful queries at the schema level.

More specifically, we formalise in Section 3 our notions of faceted interface and query, which are tailored towards RDF and OWL 2. Our notion of interface enables navigation across interconnected collections of entities, which is inherent to faceted search over RDF data. Furthermore, it abstracts from considerations specific to GUI design (e.g., facet and value ranking), while at the same time reflecting the core functionality of existing systems. Specifically, our interfaces capture both the combination of facets displayed during search and the facet values selected by users. The latter determine a *faceted query*, whose answers constitute the current results of the search. We describe such queries both as first-order logic queries satisfying certain restrictions as well as a fragment of SPARQL.

In Section 4, we study the problem of answering faceted queries over RDF graphs and ontologies captured by the OWL 2 profiles [32]—language fragments with favourable computational properties that are sufficiently powerful to capture the ontologies underpinning most existing knowledge graphs. For each of these profiles we establish tight complexity bounds and propose query answering algorithms.

In Section 5, we focus on interface generation and update. Existing techniques for RDF are based on exploration of the underlying RDF graph. We lift this approach by proposing a graph-based representation of OWL 2 ontologies and their logical entailments for the purpose of faceted navigation, which we refer to as a *facet graph*. Then, we characterise what it means for an interface to conform to an ontology, in the sense that every facet and facet value in the interface is justified by an edge in the graph (and hence by an entailment of the ontology). Finally, we propose generic interface generation and update algorithms that rely on the information in the graph, and show tractability of these tasks for ontologies in the OWL 2 profiles.

In Section 6, we present our faceted search system SemFacet and report on a proof of concept performance evaluation as well as on our practical experience with Yago.

This paper extends our conference publication [33] by providing (i) detailed proofs of our technical results; (ii) a precise account of the connection between our theoretical results in terms of first-order logic and the SPARQL standard; (iii) a detailed description of our system SemFacet; and (iv) a concrete case study based on Yago.[1]

2. Preliminaries

We use standard notions from first-order logic. We assume pairwise disjoint infinite sets of *constants* **C**, *unary predicates* **UP**, and *binary predicates* **BP**. A *signature* is a subset of $\mathbf{C} \cup \mathbf{UP} \cup \mathbf{BP}$. W.l.o.g., we assume all formulae to be rectified, that is, no variable appears free and quantified in a first-order formula φ, and every variable is quantified at most once in φ. The set of free variables of a formula φ is denoted as $\mathsf{fvar}(\varphi)$.

A *fact* is a ground relational atom and a *dataset* is a finite set of facts. A *rule* is a sentence $\forall \mathbf{x} \forall \mathbf{z}\, [\varphi(\mathbf{x}, \mathbf{z}) \rightarrow \exists \mathbf{y}\, \psi(\mathbf{x}, \mathbf{y})]$, where $\mathbf{x}$, $\mathbf{z}$, and $\mathbf{y}$ are pairwise disjoint variable tuples, the *body* $\varphi(\mathbf{x}, \mathbf{z})$ is a conjunction of atoms with variables in $\mathbf{x} \cup \mathbf{z}$, and the *head* $\exists \mathbf{y}\, \psi(\mathbf{x}, \mathbf{y})$ is an existentially quantified non-empty conjunction of atoms $\psi(\mathbf{x}, \mathbf{y})$ with variables in $\mathbf{x} \cup \mathbf{y}$. Note that we consider only rules that are *Horn* (i.e., disjunction-free), which is sufficient to capture all three profiles of OWL 2. As usual, we assume rules to be *safe*; that is, every universally quantified variable in the rule occurs in a body atom. Universal quantifiers in rules are omitted for brevity. We say that a rule is *Datalog* if its head has at most one atom and all variables are universally quantified. Finally, we define an *ontology* as a finite set of rules and facts. Note that the restriction of rule heads being non-empty ensures satisfiability of any ontology, which makes query results meaningful.

We treat $\top$ as a special symbol in **UP**, which is used to represent a tautology, and assume that any ontology with signature V mentioning $\top$ includes also the following rules:

$$\begin{aligned} A(x) &\rightarrow \top(x) && \text{for each } A \in \mathbf{UP} \cap V, \\ R(x, y) &\rightarrow \top(z) && \text{for each } z \in \{x, y\} \text{ and } R \in \mathbf{BP} \cap V. \end{aligned}$$

This treatment of $\top$ allows us to ensure safety of rules obtained from OWL 2 ontologies. Similarly, we treat equality $\approx$ as an ordinary predicate in **BP**, and assume that any ontology with signature V mentioning equality contains the following rules axiomatising its meaning:

$$\begin{aligned} x \approx y &\rightarrow y \approx x, && \\ x \approx y \wedge y \approx z &\rightarrow x \approx z, && \\ R(x, y) &\rightarrow z \approx z && \text{for all } z \in \{x, y\}, R \in \mathbf{BP} \cap V, \\ A(x) &\rightarrow x \approx x && \text{for all } A \in \mathbf{UP} \cap V, \\ A(x) \wedge x \approx y &\rightarrow A(y) && \text{for all } A \in \mathbf{UP} \cap V, \\ R(x, y) \wedge x \approx z &\rightarrow R(z, y) && \text{for all } R \in \mathbf{BP} \cap V, \\ R(x, y) \wedge y \approx z &\rightarrow R(x, z) && \text{for all } R \in \mathbf{BP} \cap V. \end{aligned}$$

OWL 2 defines three *profiles*: weaker languages with favourable computational properties [32]. Each profile ontology can be normalised as rules and facts using the correspondence of OWL 2 and first-order logic and a variant of the structural transformation.[2] An ontology where all rules are of the form given in Table 1 is

- RL if it does not contain rules (3);
- EL if it does not contain rules (1), (9), and (13); and

[1] Some of the material in this paper has also been presented at workshops without formal proceedings [34–36]; a preliminary version of SemFacet was presented as a poster [37] and a short demo paper [38].

[2] Note that the profiles provide the special concept $\perp$, which is immaterial to query answering over satisfiable profile ontologies.