On Faceted Search over Knowledge Bases [⋆]

Bernardo Cuenca Grau[1], Evgeny Kharlamov[1], Dmitriy Zheleznyakov[1], Marcelo Arenas[2], and Šarūnas Marciuška[1]

[1] University of Oxford; [2] Pontificia Universidad Católica de Chile

Motivation An increasing number of applications rely on RDF [2] and SPARQL 1.1 [3] for storing, publishing, and querying semistructured data. The functionality of many such applications is enhanced with OWL 2 ontologies [1], which are used to provide a conceptual layer on top of data and enrich query answers with implicit information.

Although the growing popularity of RDF, OWL 2, and SPARQL 1.1 has been accompanied by the development of better and better query answering engines, writing SPARQL 1.1 queries is not well-suited for the majority of users. Thus, an important challenge is the development of simple yet powerful query interfaces that capture well-defined fragments of SPARQL 1.1.

Faceted search is a prominent approach for accessing document collections that allows users to narrow down search results by incrementally applying filters, called *facets*, on the annotations associated to documents [18]. Faceted search has become a mainstream commercial technology, and it is ubiquitous in e-commerce websites. For example, hotel booking websites such as *Booking.com* allow users to refine search results by selecting suitable values in facets such as 'Price', 'Star Rating', or 'Facilities'.

Faceted search has been proposed as a suitable paradigm for querying document collections annotated with RDF, and several RDF-based faceted search systems have been developed [4, 6, 8, 13, 16, 9–11, 14, 12]. Existing approaches are, however, rather systems-oriented, and there is a lack of rigorous theoretical underpinnings. Furthermore, existing works have focused mostly on RDF, thus essentially disregarding the role of OWL 2 ontologies. In particular, the following key questions have not been satisfactorily addressed in the literature:

(Q1) What fragments of SPARQL 1.1 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 ontology?

Our goal is to provide such solid foundations. We have formalised faceted interfaces tailored towards graph-based data models, and identified a fragment of first-order logic capturing the underlying queries. We have studied the complexity of answering such queries for RDF and ontologies expressed in the OWL 2 profiles. Moreover, we have devised practical and generic algorithms for recomputing faceted interfaces in response to user actions. Finally, we have implemented and tested our faceted search algorithms

[⋆] Work supported by the Royal Society, the EPSRC projects Score!, Exoda, and MaSI[3], and the FP7 project OPTIQUE.

in a prototype system [7], with encouraging results. In this extended abstract, we provide a short overview of the main ideas underlying our approach.

Technical Approach To illustrate our definitions and make our discussion concrete, we consider an excerpt of Yago [17] about US presidents, which is depicted in Figure 1. In this figure, d_{th} and d_{bi} stand for T. Roosevelt and B. Clinton, which are annotated as US presidents with 'USpres'. Moreover, Theodore's son Kermit d_{ke} and Bill's daughter Chelsea d_{ch} are categorised as 'person'. Finally, Stanford d_s, Harvard d_h, and Georgetown d_g are annotated with 'university', and the USA d_{usa} with 'country', which, in turn, is annotated with its 'founding date' 1783-09-03. Our goal is to find US presidents who graduated from either Harvard or Georgetown and have a child who graduated from Stanford.

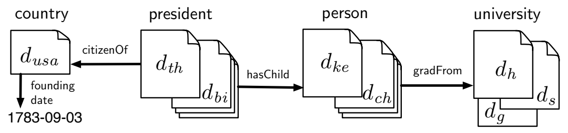

Fig. 1. Annotated entities

We model facets as pairs consisting of a predicate (or facet name) and a set of values (typically documents represented by URIs or literals). Examples of facet names are the binary relations 'gradFrom' and 'dateOfBirth', and examples of values for these facets are specific documents such as 'd_s' (Stanford) and literals such as '1858-10-27'. Selection of multiple values within a facet can be interpreted either conjunctively or disjunctively, and hence we distinguish between conjunctive and disjunctive facets. Furthermore, we distinguish a special facet type, whose values are categories (i.e., unary predicates) rather than specific documents or literals. Finally, we also allow for a special value any which denotes the set of all values compatible with the facet predicate. We assume that $\mathbf{C}$, $\mathbf{UP}$ and $\mathbf{BP}$ are pairwise disjoint infinite sets of *constants*, *unary predicates* and *binary predicates*.

Definition 1. *Let* type *and* any *be symbols not occurring in* $\mathbf{C} \cup \mathbf{UP} \cup \mathbf{BP}$. *A* facet *is a pair* $(X, \circ\Gamma)$*, with* $\circ \in \{\wedge, \vee\}$*,* Γ *a non-empty set, and either (i)* $X = \mathsf{type}$ *and* $\Gamma \subseteq \mathbf{UP}$*, or (ii)* $X \in \mathbf{BP}$*,* $\mathsf{any} \in \Gamma$ *and either* $\Gamma \subseteq \mathbf{C} \cup \{\mathsf{any}\}$ *or* $\Gamma \subseteq \mathbf{UP} \cup \{\mathsf{any}\}$. *A facet of the form* $(X, \wedge\Gamma)$ *is* conjunctive*, and a facet of the form* $(X, \vee\Gamma)$ *is* disjunctive. *In a facet* $F = (X, \circ\Gamma)$*,* X *is the* facet name*, denoted by* $F|_1$*, and* Γ *contains the* facet values *and it is denoted by* $F|_2$.

The following facets could be of use when searching over these documents.

$$F_1 = (\mathsf{type}, \vee\{\mathsf{USpres}, \mathsf{president}, \mathsf{person}, \mathsf{country}, \mathsf{university}\}),$$
$$F_2 = (\mathsf{hasChild}, \vee\{\mathsf{any}, d_{ke}, d_{ch}\}), \quad F_3 = (\mathsf{gradFrom}, \vee\{\mathsf{any}, d_h, d_s, d_g\}),$$
$$F_4 = (\mathsf{citizenOf}, \wedge\{d_{usa}, d_{uk}\}), \qquad F_5 = (\mathsf{citizenOf}, \vee\{d_{usa}, d_{uk}\}).$$

F_1 can be exploited to restrict types of entities, F_2 to narrow down search results to entities with children, where 'any' can be used to say that we are not looking for a specific child, F_3 to select an alma mater, and F_4 and F_5 to select a citizenship.

Faceted Interfaces. A faceted interface represents an arrangement of facets that can be displayed for users, and captures the choices of facet values made by them. Thus, it encodes both a query, whose answers constitute the current search results, and the possible choices of facet values available to users for further refinement. In contrast to traditional faceted search, our notion of interface allows the user to 'navigate' across interconnected sets of documents and establish independent filters to each of them.

Definition 2. *A* basic faceted interface *(BFI) is a pair* (F, Σ)*, with* F *a facet and* $\Sigma \subseteq F|_2$ *the set of* selected values*. The set of* faceted interfaces *(or interfaces, for short) is given by the following grammar, where* I_0 *and* $I_1 = (F, \Sigma)$ *are BFIs and* $F|_1 \in \mathbf{BP}$*:*

$$\begin{aligned} I &::= \mathit{path} \mid (\mathit{path} \wedge \mathit{path}) \mid (\mathit{path} \vee \mathit{path}), \\ \mathit{path} &::= I_0 \mid (I_1/I). \end{aligned}$$

A BFI encodes user choices for a specific facet, e.g., the BFI $(F_1, \{\mathsf{USpres}\})$ selects the documents categorised as US presidents. BFIs are put together in *paths*: sequences of nested facets that capture navigation between sets of documents. With nesting '/' we capture queries such as 'people with a child who graduated from Stanford' by using the interface $(F_2, \{\mathsf{any}\})/(F_3, \{d_s\})$ which first selects people having (any) children and then those children with a Stanford degree. Finally, two types of branching can be applied: $(\mathit{path}_1 \wedge \mathit{path}_2)$ indicates that search results satisfy the conditions in both path_1 and path_2, while $(\mathit{path}_1 \vee \mathit{path}_2)$ indicates that they satisfy those in path_1 or path_2. The following interface I_{ex} encodes the query 'return US presidents who graduated from Harvard or Georgetown and who have a child who graduated from Stanford'.

$$\big((F_1, \{\mathsf{USpres}\}) \wedge (F_3, \{d_h, d_g\})\big) \wedge \big((F_2, \{\mathsf{any}\})/(F_3, \{d_s\})\big).$$

Faceted Queries. Queries encoded in faceted interfaces can be captured by formulae in the positive existential fragment of first-order logic with one free variable, where in every disjunction $\varphi_1 \vee \varphi_2$, the subformulae φ_1 and φ_2 share at most one variable. Moreover, these queries involve predicates of arity at most two and are tree shaped. The output variable of a faceted query is the root variable in the query graph. We also considered extensions of faceted interfaces that allow us to choose different output variables (a functionality typically referred to as *refocusing* [5]). The query encoded by I_{ex} returns two presidents: Roosevelt and Clinton. We investigated combined complexity of faceted query evaluation for ontologies expressed in the OWL 2 profiles [15]. We showed that the problem is PTIME-complete for RL and EL ontologies and NP-complete for QL. We also investigated combined complexity under the active domain semantics (the default in SPARQL query evaluation), and showed that it is tractable for all the profiles.

Interface Generation and Update. Faceted navigation is an interactive process. Starting with an initial interface generated from a keyword search, users 'tick' or 'untick' facet values and the system reacts by updating both search results (query answers) and facets available for further navigation. We formally captured this interaction and proposed interface generation and update algorithms that are 'guided' by the (explicit and implicit) information in the ontology. Our algorithms are based on the same general principle: each element of the initial interface (resp. each change in an interface as a response of an action) must be 'justified' by a suitable entailment in $\mathcal{O}$. In this way, by exploring the ontology, we can guide users in the formulation of meaningful queries.

References

1. W3C: OWL 2 Web Ontology Language. http://www.w3.org/TR/owl2-overview/
2. W3C: Resource Description Framework (RDF). http://www.w3.org/RDF/
3. W3C: SPARQL 1.1 Query Language. www.w3.org/TR/sparql11-query/
4. Berners-Lee, T., Hollenbach, J., Lu, K., Presbrey, J., Prudhommeaux, E., Schraefel, M.M.C.: Tabulator Redux: Browsing and Writing Linked Data. In: LDOW (2008)
5. Clarkson, E., Navathe, S.B., Foley, J.D.: Generalized Formal Models for Faceted User Interfaces. In: JCDL. pp. 125–134 (2009)
6. Fafalios, P., Tzitzikas, Y.: X-ENS: Semantic Enrichment of Web Search Results at Real-Time. In: SIGIR (2013)
7. Grau, B.C., Kharlamov, E., Marciuska, S., Zheleznyakov, D., Arenas, M., Jimenez-Ruiz, E.: SemFacet: Semantic Faceted Search over Yago. In: WWW Demo (2014)
8. Hahn, R., Bizer, C., Sahnwaldt, C., Herta, C., Robinson, S., Bürgle, M., Düwiger, H., Scheel, U.: Faceted Wikipedia Search. In: BIS (2010)
9. Heim, P., Ziegler, J., Lohmann, S.: gFacet: A browser for the web of data. In: IMC-SSW. CEUR-WS, vol. 417 (2008)
10. Hildebrand, M., van Ossenbruggen, J., Hardman, L.: /facet: A Browser for Heterogeneous Semantic Web Repositories. In: ISWC (2006)
11. Huynh, D., Mazzocchi, S., Karger, D.R.: Piggy Bank: Experience the Semantic Web Inside Your Web Browser. J. Web Sem. 5(1) (2007)
12. Huynh, D.F., Karger, D.R.: Parallax and Companion: Set-based Browsing for the Data Web (2013)
13. Hyvönen, E., Saarela, S., Viljanen, K.: Ontogator: Combining View- and Ontology-Based Search with Semantic Browsing. In: XML Finland (2003)
14. Kobilarov, G., Dickinson, I.: Humboldt: Exploring Linked Data. In: LDOW (2008)
15. Motik, B., Cuenca Grau, B., Horrocks, I., Wu, Z., Fokoue, A., Lutz, C.: OWL 2 Web Ontology Language Profiles. W3C Recommendation (2009)
16. m.c. schraefel, Smith, D.A., Owens, A., Russell, A., Harris, C., Wilson, M.L.: The Evolving mSpace Platform: Leveraging the Semantic Web on the Trail of the Memex. In: Hypertext (2005)
17. Suchanek, F.M., Kasneci, G., Weikum, G.: Yago: A Core of Semantic Knowledge. In: WWW (2007)
18. Tunkelang, D.: Faceted Search. Synthesis Lectures on Information Concepts, Retrieval, and Services, Morgan & Claypool Publishers (2009)