Towards Semantic Faceted Search

Marcelo Arenas†
Dept. of Computer Science
PUC Chile

Bernardo Cuenca Grau†
Dept. of Computer Science
University of Oxford

Šarūnas Marciuška‡
Dept. of Computer Science
University of Oxford

Evgeny Kharlamov‡
Dept. of Computer Science
University of Oxford

Dmitriy Zheleznyakov‡
Dept. of Computer Science
University of Oxford

ABSTRACT

In this paper we present limitations of conventional faceted search in the way data, facets, and queries are modelled. We discuss how these limitations can be addressed with Semantic Web technologies such as RDF, OWL 2, and SPARQL 1.1. We also present a system, SemFacet, that is a proof-of-concept prototype of our approach implemented on top of Yago knowledge base, powered by the OWL 2 RL triple store RDFox, and the full text search engine Lucene.

1. MOTIVATION AND PROPOSAL

Faceted search is a technique for accessing document collections that combines text search and faceted navigation applied to the documents’ metadata. With faceted navigation, users can narrow down search results by incrementally applying multiple filters called facets [6]. During the last decade, faceted search has become a mainstream commercial technology, and it is ubiquitous in e-commerce websites and online libraries. Despite the numerous success stories, however, traditional faceted search models impose severe constraints in the way (i) faceted metadata is represented, (ii) facets are defined, and (iii) queries are formulated [4, 14]. Pushing the boundaries of faceted search beyond the current state-of-the-art requires addressing several challenges, which we discuss next. To make the discussion concrete, suppose we are looking in a travel website such as TripAdvisor for accommodation in Seoul to attend the WWW 2014 conference. We look for a 4-star or 5-star hotel with a Korean or Japanese vegetarian restaurant.

Limitations of the data model. Classical faceted search models assume that documents are not “linked” to each other. We can start our search in TripAdvisor by filling in an initial form to obtain all available hotel documents in Seoul during the conference dates. The search can then be further refined by using the facets “hotel class” and “amenities” to select 4-star or 5-star hotels with restaurants. To complete our query, we need additional constraints about restaurant documents; however, the relevant facets are associated to restaurants, and not to hotels. Thus, we switch to the interface for restaurants, where we can use the available facets to select Japanese or Korean vegetarian-friendly restaurants in Seoul. Although the hotel-specific and the restaurant-specific “views” have in common the information provided in the initial search form (i.e., city and dates), there is no link between hotel and restaurant documents and hence the constraints we imposed to restaurants are not transferred to the hotel view. Thus, although many hotels featured in TripAdvisor satisfy our query, narrowing down the search to only those hotels requires significant manual browsing effort.

Limitations of the facet model. In their most basic form, facets consist of a heading and a set of values; e.g., hotel star ratings in TripAdvisor are modelled as a facet having one value for each 1-star to 5-star rating. Many applications, however, also define facets that are hierarchical. For example, accommodation in TripAdvisor is divided into hotels, B&B, and rentals; hotels into luxury, business etc. Hierarchical facets provide background domain knowledge which can be exploited to improve faceted search; however, they are still rather limited. Although a hierarchical facet establishes dependencies between its values, the underlying semantic relationship (e.g., “is-a”, “part-of”) is undefined. There are also issues concerning dependencies between facets, which cannot be represented in such a simple model. E.g., the type of hotel and the star ratings are correlated (e.g., motels cannot be 5-star); these dependencies are typically implemented ad-hoc, which negatively impacts systems’ maintainability, performance, and reliability.

Limitations of the query model. The limitations above affect queries that users can pose. In particular, facet values for different kinds of documents cannot be joined in a single query. Thus, in TripAdvisor our example query cannot be formulated: even if we can query for both hotels or restaurants independently, when we “switch view” from hotels to restaurants, the constraints imposed on hotels are lost. Similar limitations were observed for faceted search over interlinked documents [3, 14], webpages [12], databases [7], dataspaces [17], and knowledge bases [3, 12]. Orthogonally, there are issues with the meaning of queries, which affect the way they are processed in the backend and their results are interpreted by users. In a faceted search front-end, users are presented with facets and allowed to make a multiple choice within each facet. Typically, choices in one facet are understood as logical OR, and constraints for different facets are combined with logical AND. Thus, if a user chooses “2-star” and “3-star”, they are looking for hotels with two or three stars. Multiple choice in a single facet could also be interpreted conjunctively, e.g., when the users chooses “WiFi” and “parking” facilities. Ambiguity is resolved in the backend when queries are translated into operations over inverted indices. This process is application dependent, and it is not grounded on a formal query model that can be independently studied.

Semantic Faceted Search. RDF has been proposed by many authors as a promising technology to overcome some of the limi-
Query refinement is then an iterative process, where users can either add words, which leads to a set of initial answers and initial facets. This helps in addressing sparsity of annotations and modelling of hierarchical facets. Moreover, OWL 2 axioms help in avoiding “dead ends” (i.e., facet value selections that lead to queries with the empty answer). In conventional faceted search applications, the detection of such dead ends is data driven, in the sense that the interface does not display facet values for which no document exists. Axioms provide an alternative, declarative, way to detect dead ends during faceted search, e.g., by exploiting axioms expressing disjointness between classes of objects.

SemFacet is available as a Web service [1] and runs on a machine with 1vCPU, 4Gb of memory, and 20Gb of disk space. Although we have not formally evaluated our system, preliminary experiments show typical response time comparable with well known conventional faceted search systems.

3. REFERENCES