Chapter 10

Searching Linked Objects with Falcons: Approach, Implementation and Evaluation

Gong Cheng
Nanjing University, China

Yuzhong Qu
Nanjing University, China

ABSTRACT

The rapid development of the data Web is accompanied by increasing information needs from ordinary Web users for searching objects and their relations. To meet the challenge, this chapter presents Falcons Object Search, a keyword-based search engine for linked objects. To support various user needs expressed via keyword queries, for each object an extensive virtual document is indexed, which consists of not only associated literals but also the textual descriptions of associated links and linked objects. The resulting objects are ranked according to a combination of their relevance to the query and their popularity. For each resulting object, a query-relevant structured snippet is provided to show the associated literals and linked objects matched with the query for reflecting query relevance and even directly answering the question behind the query. To exploit ontological semantics for more precise search results, the type information of objects is leveraged to support class-based query refinement, and Web-scale class-inclusion reasoning is performed to discover implicit type information. Further, a subclass recommendation technique is proposed to allow users navigate class hierarchies for incremental results filtering. A task-based experiment demonstrates the promising features of the system.

DOI: 10.4018/978-1-60960-593-3.ch010
SEARCHING LINKED OBJECTS WITH FALCONS

Approach, Implementation and Evaluation

The Web is developing toward the era of data Web, or the Semantic Web. In recent years, substantial RDF data describing objects and their relations in various domains has been published as linked data on the Web. More importantly, these data sets have been interlinked, leading to a Web of data, grounded on which a lot of interesting applications have been developed.

Meanwhile, as before, search is one of the most common activities in daily life. People use text-based Web search engines at all times. Thus naturally, a key feature of the emerging data Web that would benefit ordinary Web users is to assist them in finding more accurate information on the Web in a shorter time. Recall that on the hypertext Web, people who seek information have to firstly retrieve Web documents, and then look through the texts for the desired knowledge by themselves. By contrast, on the data Web, knowledge has been represented in a structured manner so that it is possible to answer the question behind a query more efficiently. For example, previously in order to find relations between two people, we usually combine their names into a single keyword query and submit it to a text-based Web search engine. Then, we manually locate the two names in the text of each resulting webpage, read their contexts, and finally conclude potential relations from texts. Evidently, the entire process is rather time-consuming. Differently, linked data exactly describes the attributes of and the relations between objects. Thereby, a search engine that utilizes such data having well-defined meanings may automatically find and present accurately defined relations between two people.

Many novel search engines have been developed for the data Web. Most of these systems focus on RDF document search (d’Aquín, Baldassarre, Gridinoc, Sabou, Angeletou, & Motta, 2007; Oren, Delbru, Catasta, Cyganiak, Stenzhorn, & Tummarello, 2008) or ontology search (Ding, Pan, Finin, Joshi, Peng, & Kolari, 2005). Recall that an RDF document serializes an RDF graph; an ontology, as a schema on the data Web, defines classes and properties for describing objects. Although both RDF document search and ontology search are essential for application developers, they can hardly serve ordinary Web users directly. Instead, object-level search is in demand and dominates all other Web queries (Pound, Mika, & Zaragoza, 2010).

To meet the challenge, we present our solution called Falcons Object Search (http://ws.nj.edu.cn/falcons/objectsearch/), which firstly is a keyword-based object search engine. For each discovered object, the system constructs an extensive virtual document consisting of textual descriptions extracted from its concise RDF description. Then an inverted index is built from terms in virtual documents to objects for supporting basic keyword-based search. That is, when a keyword query arrives, based on the inverted index, the system matches the terms in the query with the virtual documents of objects to generate a result set. The resulting objects are ranked in terms of both their relevance to the query and their popularity. For each resulting object, the system computes a query-relevant structured snippet to show the associated literals and linked objects matched with the query. Thereby, the extensiveness of virtual documents and the structural nature of snippets make the system go beyond searching for a particular object. For example, users can search for objects having a certain property, or can submit keyword queries describing two or more objects to seek their relations. Besides, the type information of objects, expanded by class-inclusion reasoning, is used to provide class-based query refinement. A technique of recommending subclasses is implemented to allow navigating class hierarchies for incremental results filtering.