Scripting challenge entry: A lookup index for Semantic Web resources

Eyal Oren and Giovanni Tummarello

Digital Enterprise Research Institute
National University of Ireland, Galway
Galway, Ireland

Finding Semantic Web statements The Semantic Web can be seen as a large knowledge-base of statements about resources, forming an interconnected graph through multiple references to the same resources. But the graphs in the Semantic Web are decentralised: there is not one single knowledge base that contains the graph of statements but instead anyone can contribute statements on his "personal" web-space. The complete graph is only visible after crawling and integrating the fragments mentioned on these personal subspaces. For developers of Semantic Web applications, which operate on Semantic Web data, the decentralisation poses a challenge: how and where to find statements (information) about certain resources?

A simple lookup index We have developed (an initial version of) a simple lookup index called Sindice that helps developers in this situation. Sindice is available online[1], the code is available open-source[2] under the LGPL license.

Benefits for the user By itself our index has no benefits for the end-user. But any application that uses Semantic Web data, such as the Disgo[3], Tabulator[4], or BrowseRDF[5] RDF browsers, the DBin[6] information sharing client, or the SIOC browser[7], can use Sindice to offer *their* users a better service. Next to every resource that they encounter these applications can place a "find more information". When users click that button, their application would request a list of relevant sources with more information about a resource, and follow one or more of these sources, learning things that others said about that particular resource. Instead of crawling the Semantic Web themselves, they need only ask Sindice for pointers to good sources.

[1] http://activerdf.org/sindice
[2] http://launchpad.net/sindice
[3] http://sites.wiwiss.fu-berlin.de/suhl/bizer/ng4j/disco/
[4] http://www.w3.org/2005/ajar/tab
[5] http://browserdf.com
[6] http://dbin.org
[7] http://activerdf.org/sioc

Crawling sources Users can request explicit crawls of their updated documents through the RESTful API[8]. Apart from that, Sindice periodically (currently every twenty minutes) requests recently updated documents itself from http://pingthesemanticweb.com and visits each source to index its contents. Once our hardware infrastructure is stable we will also index large RDF dumps from Swoogle[9], SWSE[10] and Wikipedia[11]. With our current datastructure, and based on our currently indexed data, we estimate that storing 300 million resources (which is our minimal target) will occupy approximately 77Gb of data, which is quite readily available on ordinary hardware.

Ranking results We rank results using a cheap and potentially "useful enough" algorithm. We give precedence to sources on the same hostname as the queried resource (following the linked-data model that resources should have meaningful descriptions on their own de-referenceable URIs. We further rank sources on their PageRank, their size (in amount of statements) and their amount of statements. We also investigate a feedback model where often-used sources would be boosted as well.x

Project scope Sindice is a simple lookup index and its scope is limited: it does not index the actual RDF in the documents but only the resources mentioned; it does not allow full queries on the data but only lookups from resource to mentioning sources. Keeping the scope focused we are able to keep our index small and fast, namely an on-disk hashtable: `resource` ⇒ `source[]`.

Implementation Sindice is built using the Ruby scripting language for rapid prototyping and uses the Ruby on Rails Web application framework to handle routing of requests and offering a RESTful API with various response formats (HTML, JSON, and XML). Sindice uses several external libraries such as the Redland[12] "rapper" RDF parser and the QDBM[13] persistent hashtable. Sindice also uses several online services such as http://pingthesemanticweb.com, which is queried periodically to find recent RDF documents, and an estimation[14] of Google's PageRank to estimate relative importance of sources. Apart from the automatically generated Ruby on Rails code, the core Sindice library is implemented in around 200 lines of code (including comments) and the Web application, handling requests and generating HTML, JSON, and XML responses in around 130 lines of code (including comments).

[8] http://activerdf.org/sindice/parse/URL
[9] http://swoogle.umbc.edu
[10] http://swse.deri.org
[11] http://dbpedia.org
[12] http://librdf.org
[13] http://qdbm.sourceforge.net
[14] http://seopen.com/seopen-tools/pagerank.php