QueryMed: For Querying Biomedical Data on the Web*

Oshani Seneviratne
Massachusetts Institute of Technology
Cambridge, MA
USA
oshani@csail.mit.edu

Rachel Sealfon
Massachusetts Institute of Technology
Cambridge, MA
USA
rsealfon@csail.mit.edu

ABSTRACT

QueryMed is a query builder and result set visualizer for biomedical data that allows users to easily construct and run translational medicine queries across multiple data sources. The system is accessible for users who are unfamiliar with the SPARQL query language [8] or the structure of the underlying ontologies. It permits users to retrieve data from an information-rich default set of resources or to draw on their specialized domain knowledge to determine the most appropriate data sources to query. User input is provided through an intuitive interface, translated into SPARQL queries, and executed at the relevant endpoints. The results are presented in an accessible interface that allows query refinement and filtering.

Keywords

Biomedical Ontologies, SPARQL, Query Federation, Query Building, Semantic Web, User Interfaces

1. INTRODUCTION

The quantity of publicly available data in the biomedical domain has dramatically increased in recent years. With the linked open data movement, the Semantic Web community has been very proactive in converting these rich information data sources to Resource Description Framework (RDF) triplestores [2, 10]. However, to exploit the abundance of biomedical data on the Semantic Web, there is a need for easy-to-use systems that do not require the end user to have knowledge of the underlying structure of the data or of the ontologies used in describing the data. These systems should support queries that run across multiple datasources. There is also a need for efficient hybrid interfaces that allow browsing data resources as well as performing queries [7].

*Interested readers are advised to visit http://code.google.com/p/querymed to learn more about QueryMed.

2. OVERVIEW

The QueryMed system is designed to allow users without specialized knowledge of query languages to browse and query the extensive biomedical data resources available in RDF triplestores. The simple user interface of QueryMed is intuitive for users with no knowledge of SPARQL and allows the user to run queries across multiple biomedical datasources. Queries can be run against a default list of SPARQL endpoints, or against a set of user-defined endpoints. The system automatically translates the user input into a SPARQL query for each individual endpoint, executes the query, combines the results, and returns them to the user. The user can choose to refine the query by iteratively modifying the original search terms and by filtering the result set. The system's advanced query functionality enables the user to construct complex logical SPARQL queries that take advantage of the underlying structure of the data. A general overview of QueryMed architecture is shown in Figure 1.

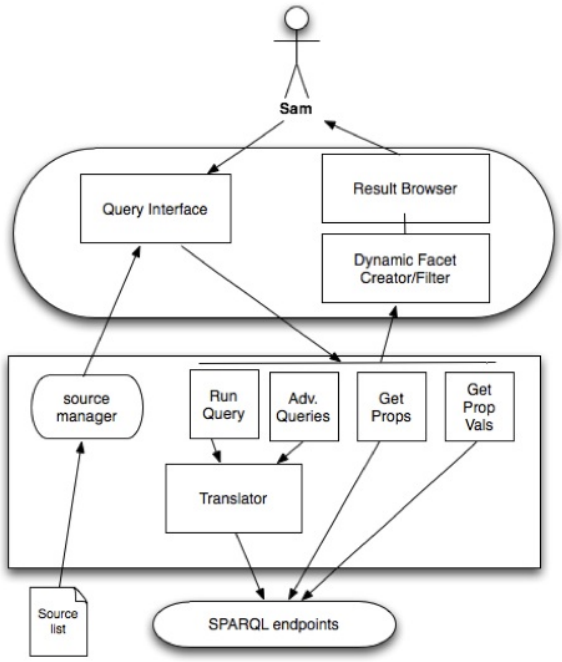

Figure 1: QueryMed Architecture

2.1 Sample Use Case

Imagine a scenario in which a physician, Sam, is interested in locating freely available resources related to coronary artery disease. Using QueryMed, Sam first runs a basic search by entering "coronary artery disease" in the search box. He sees a table displaying disease names in the Diseasome database and drugs in DailyMed and Drugbank that relate to coronary artery disease. He can then filter the results using additional search terms. For example, he knows that the route of administration of the drug that he is looking for is injection, so he filters the drug query results on the route of administration field using the query term "injection." He is now interested in finding relevant clinical trials for his patient. The clinical trial database (LinkedCT [1]) is not in the default set of endpoints, so he selects the "Refine Query" option to choose additional endpoints to search. He sees a list of default endpoints, and selects the "Add" option to include an additional endpoint. After entering the name and URL of the LinkedCT endpoint, he is able to search for clinical trials for which his patient may be eligible. He is also interested in further refining the search, so he uses the advanced search option to search the data available in the Diseasome endpoint for a list of diseases whose class is "Cardiovascular" or for which the associated gene is "ABCA1." Using the QueryMed advanced search interface, the complex SPARQL query corresponding to his question is automatically constructed, and he can view the query results conveniently displayed in a table. A demo illustrating this scenario is available at `http://dig.csail.mit.edu/2010/Papers/www-ws-colab-science/videos/querymed-demo.mov`.

3. RELATED WORK

A number of existing tools aim to provide a user-friendly interface for browsing Semantic Web data, or to allow users to perform federated queries. The SMART query tool [4] is a Web-based application designed to allow biologists to run queries written in Manchester OWL syntax. GoWeb [5] allows users to perform a hybrid search, running keyword-based queries and then filtering based on ontological concepts. BioGateway [3] provides a Web interface to query a single provided SPARQL endpoint that includes graphs from several biomedical resources. Another query-building tool, Twinkle [6], offers a stand-alone graphical user interface to load and edit SPARQL queries. The DARQ system [9] is designed to allow the user to run integrated queries against multiple SPARQL endpoints. But it does not offer a graphical user interface to facilitate use by biomedical domain experts who are not familiar with SPARQL query syntax.

Features of the QueryMed system that distinguish it from similar systems include the ability both to perform keyword queries and to construct more advanced queries taking advantage of the structure of the data, a hybrid interface that enables the user to both query and browse data, and a graphical user interface permitting the dynamic addition of endpoints. Furthermore, the QueryMed user interface is attractive, easy to interact with, and capable of handling a flexible range of user input. Another unique feature of our system is the property-based advanced query interface. This interface enables users to take advantage of the structures of the underlying ontologies used to represent the data without prior knowledge of these structures.

4. CONCLUSION

The main contributions of QueryMed are: dynamic construction of complex SPARQL queries based on intuitive user input; dynamic addition of user-specified datasources; and ability to run queries over multiple datasources. Because the system is flexible and easy to use, we believe that it will be valuable to the biomedical community. We also believe that developing systems such as such as QueryMed, which make SPARQL endpoints easily accessible to end-users, will entice more people to expose their biomedical data as linked open data, thus promoting the growth of the Linked Open Data cloud.

5. REFERENCES

[1] Linkedct, http://linkedct.org/sparql.

[2] W3c sweo community project, linking open data.

[3] E. Antezana, W. Blondé, M. Egana, A. Rutherford, R. Stevens, B. De Baets, V. Mironov, and M. Kuiper. Structuring the life science resourceome for semantic systems biology: lessons from the BioGateway Project. *PSWAT4LSÑBurger A, Paschke A, Romano, et al, eds*, 435, 2008.

[4] A. D. L. Battista, N. Villanueva-Rosales, M. Palenychka, and M. Dumontier. Smart: A web-based, ontology-driven, semantic web query answering application. In *Semantic Web Challenge*, 2007.

[5] H. Dietze and M. Schroeder. Goweb: a semantic search engine for the life science web. *BMC Bioinformatics*, 10 Suppl 10, 2009.

[6] L. Dodds. Twinkle: A sparql query tool, http://www.ldodds.com/projects/twinkle/.

[7] A. Jentzsch, B. Andersson, O. Hassanzadeh, S. Stephens, and C. Bizer. Enabling tailored therapeutics with linked data. In *World Wide Web Conference: Linked Data On the Web Workshop*, 2009.

[8] E. Prud'hommeaux and A. Seaborne. Sparql query language for rdf. w3c recommendation. 2008.

[9] B. Quilitz and U. Leser. Querying distributed rdf data sources with sparql. In *ESWC*, pages 524–538, 2008.

[10] Y. Yip. Accelerating knowledge discovery through community data sharing and integration. *Yearb. Med Inform.*, 2009.