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 end-users to easily construct and run translational medicine queries across multiple data sources. It allows users to select the data sources they wish to use, drawing on their specialized domain knowledge to decide the most appropriate data sources to query. User input is translated into SPARQL [13] queries and executed at the relevant endpoints. The results are presented in an intuitive user interface that allows query refinement and filtering.

Keywords

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

1. INTRODUCTION

Answering many medically and biologically relevant questions with information available on the Web requires searching, filtering, and combining information from multiple data sources. For example, a physician who knows a patient's symptoms, current medications, and genotype may wish to determine the treatment plan and identify clinical trials for which the patient is eligible for. However, there is no single data source that the physician can use to figure out the answers. The information that the physician needs must be gathered from numerous data sources such as Pubmed, DailyMed, Drugbank, LinkedCT, Diseasome, and Gene Ontology [6, 1, 3, 5, 2, 4]. The quantity of publicly available data in the biomedical domain has dramatically increased over the recent years. With the linked open data movement, the Semantic Web community has been very proactive in converting these rich information resources to RDF [7]. The biomedical domain is in fact among the early successes of the Semantic Web, due to the rapidity with which the community has made its data available in RDF triple stores [15].

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

However, to exploit the abundance of this data on the 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 the ontologies used in describing the data. These systems should also support the queries to be run across multiple data sources. There is also a need for efficient hybrid interfaces that allow both browsing and querying of data [12].

2. OVERVIEW

QueryMed allows querying of multiple biomedical data sources very easily. 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. If the user is not happy with the results, the query can be refined by iteratively modifying the original terms used in the query, and by filtering the result set. The advanced query functionality of QueryMed enable the user to easily construct complex logical SPARQL queries that take advantage of the underlying structure of the data. The simple user interface of QueryMed is designed to be intuitive for users with no knowledge of SPARQL. A general overview of QueryMed architecture is shown in Figure 1.

2.1 QueryMed in Action

Imagine a scenario in which a physician called Sam, is interested in finding freely available resources related to coronary artery disease. Using QueryMed Sam first tries a basic search over all default resources, by entering "coronary artery disease" in the search box. Sam then sees a list of disease names in the Diseasome database and drugs in DailyMed and Drugbank that relate to coronary artery disease displayed in a table. 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 then prints the table of results. he is now interested in finding relevant clinical trials for her patient. However, the clinical trial database (LinkedCT [5]) 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 her patient may be eligible. He is also

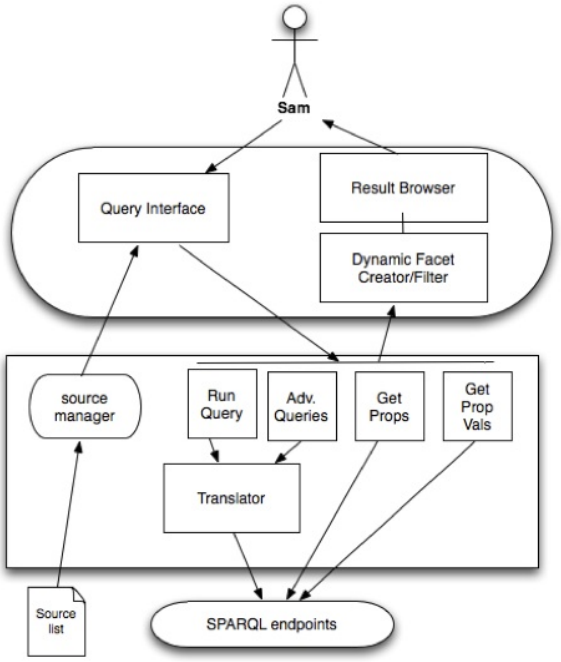

Figure 1: QueryMed Architecture

interested in further refining her search, so he uses the advanced search option to construct a query that takes advantage of the underlying structure of the RDF data in the database. He searches 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 describing 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 [9] is a Web-based application designed to allow biologists to run queries written in Manchester OWL syntax. Unlike in SMART, QueryMed allows constructing queries intuitively by specifying keywords for user selected properties, to iteratively refine query results, and to dynamically add additional endpoints. GoWeb [10] allows users to perform a hybrid search, running keyword-based queries and then filtering based on ontological concepts. However, GoWeb functions as a search engine that has in-built ontological background knowledge to improve the search results, whereas QueryMed utilizes user input in constructing SPARQL queries without any ontological backing. BioGateway [8] provides a Web interface to query a provided single SPARQL endpoint that includes graphs from several biomedical resources. However, BioGateway does not facilitate dynamic addition of endpoints like in QueryMed. Another, query builder, Twinkle [11], offers a stand-alone graphical user interface to load and edit SPARQL queries. Our system differs from the Twinkle, because in QueryMed, the user can provide input in the form of keywords, and has the option to restrict the query if she wishes to run a more precise query. Most SPARQL query engines are designed to run queries against individual endpoints. But it is often useful to draw on multiple web resources in answering a query. The DARQ system [14] is designed to allow the user to run integrated queries against multiple SPARQL endpoints. But it does not offer a graphical user interface as in QueryMed to facilitate use by biomedical domain experts who are not familiar with SPARQL query syntax.

4. CONCLUSION

The main contributions of QueryMed are: dynamic construction of complex SPARQL queries based on intuitive user input; dynamic addition of user-specified endpoints; and ability to run queries over multiple endpoints. Because it is flexible and easy to use, we believe that it will be of immense value 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 data as linked open data thus promoting the growth of the LODD cloud.

5. REFERENCES

[1] Dailymed, http://dailymed.nlm.nih.gov/dailymed/.
[2] Diseasome, http://www4.wiwiss.fu-berlin.de/diseasome/.
[3] Drugbank, http://www.drugbank.ca.
[4] Gene ontology, http://www.geneontology.org.
[5] Linkedct, http://linkedct.org/sparql.
[6] Pubmed, http://pubmed.bio2rdf.org/sparql/.
[7] W3c sweo community project, linking open data.
[8] 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.
[9] 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.
[10] H. Dietze and M. Schroeder. Goweb: a semantic search engine for the life science web. *BMC Bioinformatics*, 10 Suppl 10, 2009.
[11] L. Dodds. Twinkle: A sparql query tool, http://www.ldodds.com/projects/twinkle/.
[12] 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.
[13] E. Prud'hommeaux and A. Seaborne. Sparql query language for rdf. w3c recommendation. 2008.
[14] B. Quilitz and U. Leser. Querying distributed rdf data sources with sparql. In *ESWC*, pages 524–538, 2008.
[15] Y. Yip. Accelerating knowledge discovery through community data sharing and integration. *Yearb. Med Inform.*, 2009.