QueryMed: An Intuitive Query Builder for Biomedical Data available 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 SPARQL 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 a SPARQL query or multiple queries and executed at the relevant SPARQL endpoints for the results.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous

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 over 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 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 [16]. However, to allow end users to exploit the abundance of biomedical RDF data, 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, and that also allow them to run federated queries across multiple data sources. There is also a need for efficient hybrid interfaces that allow both browsing and querying data [13].

Answering many medically and biologically relevant questions requires searching, filtering, and combining information from multiple data sources. For example, a physician may know her patient's personal information, symptoms, current medications, and genotype. She may wish to determine the patient's treatment plan and identify clinical trials for which the patient is eligible. Although the physician has a single question–"based on the information I have about this patient, what is the best treatment plan and set of clinical trials available?"–there is no single data source that the physician can use to answer this question. 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]. Her question must be broken up into discrete pieces that can be executed individually at one data source at a time. Since the physician must search many data sources in order to find an answer to her single question, she requires a system that can automatically run queries over multiple data sources. Also, the physician may not know SPARQL syntax, the location of the SPARQL endpoints (i.e. data sources), or the structure of the ontologies used to describe the data in the endpoints. She is likely to want an intuitive way to query and to display the query results. Developing intuitive ways to query multiple SPARQL endpoints and to display results is both an important and a challenging problem. Our system, QueryMed, allows users with no knowledge of the SPARQL query language or the structure of the underlying ontologies to easily run queries across multiple SPARQL endpoints.

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-selected endpoints. Additional endpoints can be selected in order to utilize resources that are not included in the default list. 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 ontologies. 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 **??**.

3. QUERYMED IN ACTION

Imagine a scenario in which someone, let's say a physician, is interested in finding freely available resources related to coronary artery disease. She first tries a basic search over all default resources, by entering "coronary artery disease" in the input search box. She 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. She can then filter the results using additional search terms. For example, she knows that the route of administration of the drug that she is looking for is injection, so she filters the drug query results on the route of administration field using the query term "injection." She then prints the table of results. She is now interested in finding relevant clinical trials for her patient. However, the clinical trial database LinkedCT is not in the default set of endpoints, so she selects the "Refine Query" option to choose additional endpoints to search. She 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, she is able to search for clinical trials for which her patient may be eligible. She is also interested in further refining her search. She uses the advanced search option to construct a boolean query that takes advantage of the underlying structure of the RDF data in the database. She 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 her question is automatically constructed, and she can view the query results conveniently displayed in a table.

4. 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 search results, whereas QueryMed utilizes user input in constructing SPARQL queries without any ontological backing. BioGatewa [8] provides a Web interface to query a provided single SPARQL endpoint that includes graphs from several biomedical resources. BioGateway does not facilitate dynamically 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 system in several aspects. First of all, in Twinkle, the user is expected to know what is already available at the SPARQL endpoints to write the query. But 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. Second, although Twinkle was designed to be a more general purpose system, it only supports a small number of specific SPARQL endpoints, while QueryMed allows the user to dynamically add SPARQL endpoints. 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 to facilitate use by biomedical domain experts who are not familiar with SPARQL query syntax.

5. CONCLUSION

One of the main goals of the system was to make it accessible for users who are unfamiliar with SPARQL or the structure of the underlying ontologies. The main contributions of our system 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 our system is flexible and easy to use, we believe it will be of use 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.

6. 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] C. Goble and R. Stevens. State of the nation in data integration for bioinformatics. *J. of Biomedical Informatics*, 41(5):687–693, 2008.

[13] 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.

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

[15] M. Sharp, O. Bodenreider, and N. Wacholder. A framework for characterizing drug information sources. 2008.

[16] Y. Yip. Accelerating knowledge discovery through

community data sharing and integration. *Yearb. Med Inform.*, 2009.