

QueryMed: An Intuitive SPARQL Query Builder for Biomedical RDF Data

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

Rachel Sealon
Massachusetts Institute of Technology
Cambridge, MA
USA
rsealon@csail.mit.edu

ABSTRACT

We have developed an open-source SPARQL query builder and result set visualizer for biomedical data, QueryMed, that allows end users to easily construct and run translational medicine queries across multiple data sources.

QueryMed allows a flexible range of queries relevant to a wide range of biomedical topics and runs queries across multiple SPARQL endpoints. The system is accessible for users who are unfamiliar with the SPARQL query language or the structure of the underlying ontologies. The system allows users to select the data sources that they wish to use, drawing on their specialized domain knowledge to decide the most appropriate data sources to query. Users can add additional data sources if they are interested in querying endpoints that are not in the default list. The system automatically translates the user input into a SPARQL query for each individual endpoint, combines the results, and returns them to the user. After retrieval of the initial result set, query results can be improved by iteratively modifying the original query terms, and by filtering the result list. The advanced query functionality of the system allows the user to easily construct complex logical SPARQL queries that exploit the underlying structure of the RDF data.

Keywords

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

1. QUERYMED OVERVIEW

1.1 QueryMed Overview

A general overview of QueryMed architecture is shown in Figure ???. The main components of the system are the user interface and the proxy server that takes input from the user and retrieves relevant biomedical data from remote SPARQL endpoints. After a user submits a query from the user interface, the query is translated by the proxy server into individual SPARQL queries for each remote endpoint. The query results are returned from the remote endpoints, combined by the proxy server, and presented to the user.

1.1.1 User Interface

The QueryMed user interface is designed to be intuitive for the end user, yet flexible enough to permit a broad range

Copyright is held by the author/owner(s).

WWW2010, April 26-30, 2010, Raleigh, North Carolina.

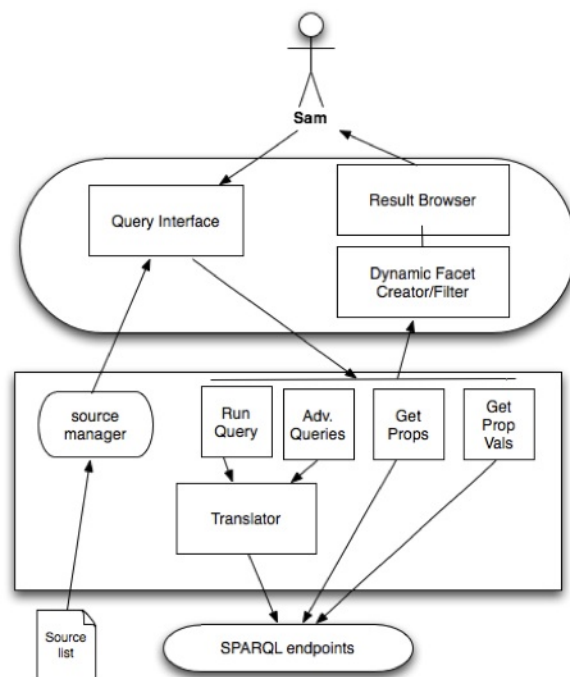


Figure 1: QueryMed Architecture

of interesting queries. The basic query interface allows the user to run simple queries, and is designed for maximal ease of use (Figure 2). The advanced search capabilities allow the user to easily construct complex logical queries that take advantage of the underlying structure of the biomedical ontologies. The user interface also allows the user to iteratively refine queries, and displays the query results, which have been retrieved from multiple SPARQL endpoints.

1.1.2 Proxy Server

The proxy server acts as an intermediary between the user interface and the remote SPARQL endpoints. Its functionality is twofold:

1. Execute the SPARQL queries at the relevant remote SPARQL endpoints and consolidate the results to be presented in the user interface.
2. Cache the results of the current query so that refinements of the query will have reduced network and

	Hybrid Interface (Combines Querying & Browsing)	Provides Local Caching	Queries Multiple Sources	Dynamic Addition of Sources	Allows Keyword Queries	Open Source	GUI
QueryMed	Yes	Yes	Yes	Yes	Yes	Yes	Yes
SMART	No	Yes	Yes	No	No	Yes	Yes
DARQ	No	No	Yes	N/A	No	Yes	No
GoWeb	Yes	No	Yes	No	Yes	No	Yes
BioGateway	Yes	Yes	Yes	No	No	Yes	Yes
Twinkle	No	No	Yes	No	No	Yes	Yes

Table 1: Comparison of selected features of the QueryMed system with other related systems.

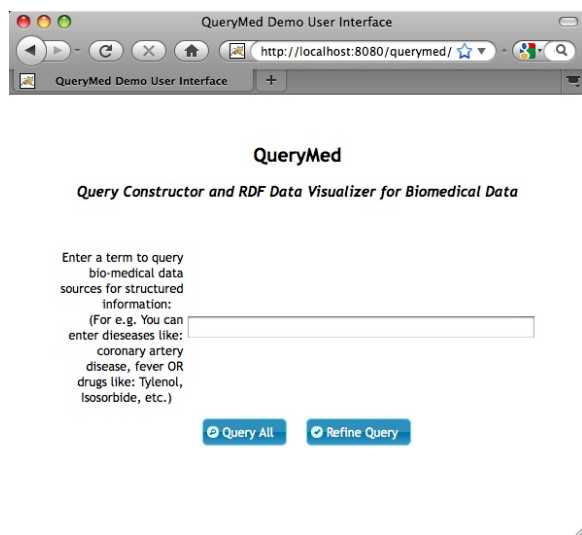


Figure 2: Initial Basic Query Interface

query execution latency.

The specific components in the Proxy Server are as follows.

- **Source Manager:** The Source Manager reads the source list and populates the default query list on the user interface. It also keeps track of the default endpoints, the currently selected endpoints, and the endpoints that have been dynamically added.
- **Translator:** The Translator is responsible for translating the user query into valid SPARQL syntax. The Translator obtains the parameters to construct the query from the input the user specifies in the Query Interface, and dynamically constructs a SPARQL query based on the user input.

1.1.3 Implementation

QueryMed is implemented in Java in the backend and JavaScript, HTML and CSS in the frontend. In the backend, the Jena library [2] is used to run the SPARQL queries and 4store [1] is used as a triple store to provide caching support. The JQuery library [3] was used to develop an attractive user interface.

1.2 QueryMed Resources

The source code for the QueryMed system is available at the QueryMed Google Code project: <http://code.google.com/p/querymed>

A video illustrating a sample use case can be found at: <http://dig.csail.mit.edu/2010/Papers/www-ws-colab-science/videos>

2. 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.

Table 1 compares selected features of the QueryMed system with other related systems. The QueryMed system was unique among the systems that we found in that it allows endpoints to be dynamically added by the user, and provides a hybrid interface that enables the users to both query and browse data. Other 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. This feature increases the ease of use of our system relative to other similar systems. Furthermore, the Javascript-based user interface of the QueryMed system, implemented using the JQuery library, makes our user interface particularly attractive, easy to interact with, and capable of handling a variety of user input events. Another unique feature of our system is the property-based advanced query interface. This interface enables users to take advantage of the structure of the underlying ontologies used to represent the data without prior knowledge of the ontology structures.

3. CONCLUSION

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 QueryMed, which make SPARQL endpoints easily accessible to end users, will entice more people to expose their data as linked open data.

4. REFERENCES

- [1] Garlik. 4store, an efficient, scalable and stable rdf database, <http://4store.org/>.

- [2] B. McBride. Jena - a semantic web framework.
- [3] J. Resig. JQuery –javascript library, <http://jquery.com>.