QueryMed :: Intelligent Query Translator and RDF Data Visualizer for Biomedical Data

Rachel Sealfon (sealfon@csail.mit.edu)
Oshani Seneviratne (oshani@csail.mit.edu)

Abstract

One of the main goals of the Semantic Web is to add semantics to the current Web in such a way that it is easier to make automated tools to reason and make useful inferences using RDF data. Significant amount of medical data has been organized into ontologies in a step towards realizing this goal [1]. Most of these ontologies are in Unified Medical Language System (UMLS) or in Web Ontology Language (OWL). Along with these ontologies, there are many bio medical data-sets that are available through SPARQL endpoints [2]. With this increased amount of data on the Web, there is a need for tools and techniques that allow physicians and patients who have no knowledge of SPARQL, or of other query languages, or of the underlying structure of the data, to effectively query, search, and utilize these data in their day-to-day work or for general interest. Because of the complexity of the underlying structure of the data, developing intuitive ways to query and display this information is both an important and a difficult problem. We propose to develop an RDF Data Visualizer and Query Translator for Biomedical Data, "QueryMed", that allows end users to easily construct and run translational medicine queries to find answers to their questions.

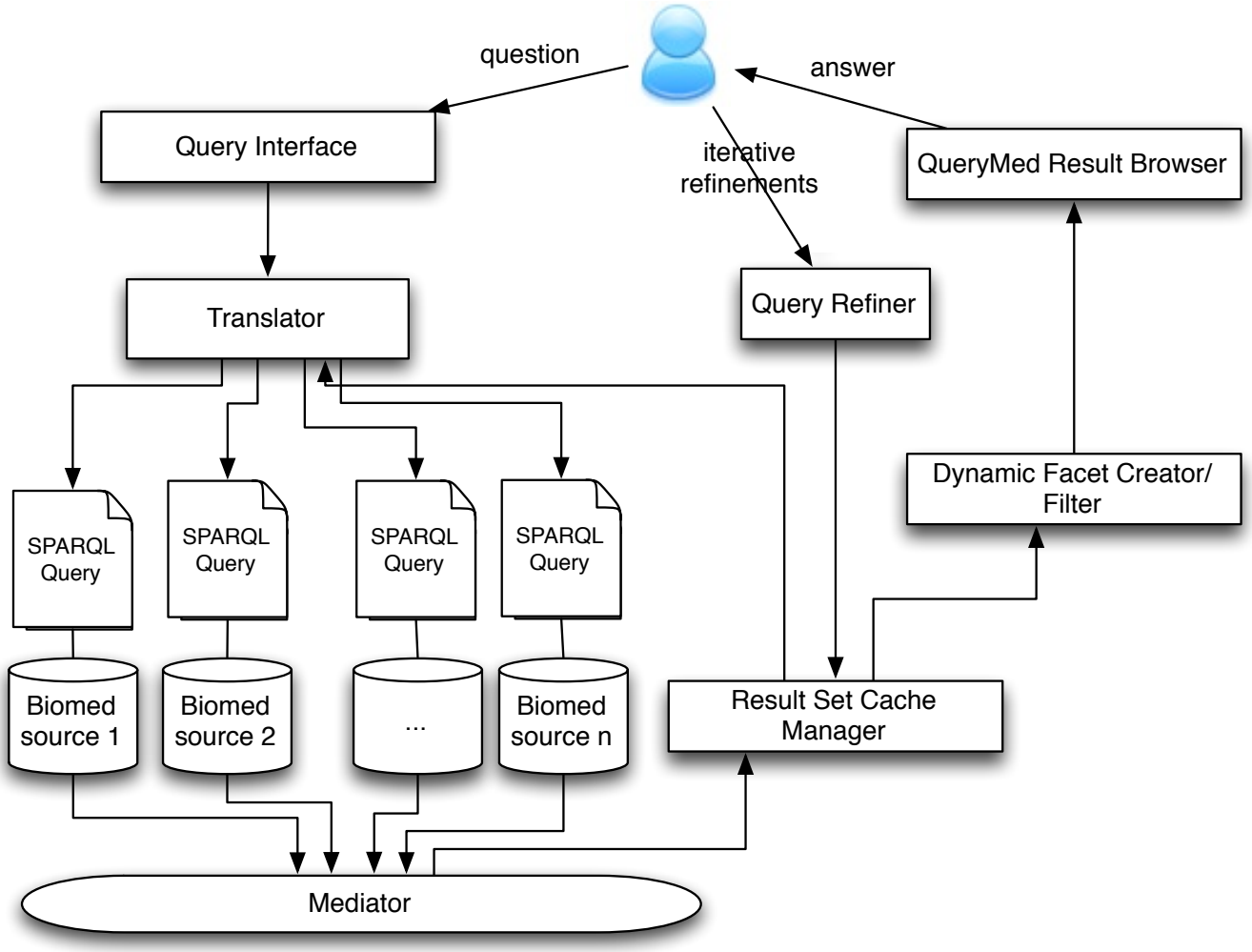
Overview

Our system can be used to intuitively search and combine information stored using multiple ontologies in several data sources. For example, a physician may know her patient's personal information, symptoms, current medications, and genotype. She may wish to figure out the patient's diagnosis and treatment plan, find any relevant papers in PubMed [3] that relate to the prevailing condition, view the function of genes that the patient has a mutation in, or identify clinical trials that the patient can be referred to. The proposed system will query remote data sources such as DailyMed [4], LinkedCT [5], Drugbank [6], Dieseasome [7], GO [8] and other relevant data sets that are publicly available.

QueryMed has to determine how to structure the query, so that the relevant part of the overall query will be executed at the corresponding data source. Once the results are received from that particular data source, it may be pipelined to another query to be executed at another data source. This process will be dynamic and it will be determined in an optimal manner. Once the relevant results are retrieved from all the applicable data sources, it will be presented to the user in an intuitive user interface that allows filtering and refining of the initial query. For example, suppose the physician has a patient with the flu, and has searched for drugs used to treat flu. Her first query result happens to be Tamiflu; she might then wish to search for PubMed articles containing the word "Tamiflu" in the title. Further, the physician

might also be interested in finding clinical trials that the patient can be directed to participate in. It should also be possible to refine the query by adding/deleting/updating any of the search parameters. All of the data sources that are queried will be at remote endpoints, and the trustworthiness of the data will be guaranteed by providing information about where the data came from. For example, the physician might want to know which resource states that Tamiflu is a possible treatment for the flu, because she trusts some resources more than others.

The proposed system architecture for QueryMed is as follows:



Query Interface: This will be the main interface the user can ask questions from.

Query Refiner: The user will be able to tweak the parameter in the initial query and ask "what if" questions.

Translator: This will be responsible for translating the user query into SPARQL. It will try to figure out the optimal query execution plan, what parts of the query should be executed at which data source,

whether query pipelining is needed and which data set should act as the intermediate result set and write the query using SPARQL syntax.

Mediator: Executes the queries on various data sources and combines them to get a unified view. Also handles the pipelining as determined by the Translator.

Result Set Cache Manager: Stores the intermediate results sets.

Dynamic Facet Creator: This will allow filtering of the result set using an Exhibit [9] like interface.

QueryMed Result Browser: The interface the user sees when browsing the QueryMed system.

Schedule

Here is a rough timeline outlining the main challenges and progress steps of this project:

1. Monday 11/9: Collect all data sources that are in a usable format, consult domain experts and verify the user requirements (i.e. figure out what type of questions they are interested in asking in the Query-Med system)
2. Monday 11/16: Implement the Translator, Mediator and the Result Set Cache Manager
3. Monday 11/23: Preliminary prototype allowing some basic queries
4. Monday 11/30: Implement Query Refiner, Dynamic Facet Creator and refine the UI
5. Monday 12/7: Prepare Documentation and Write the Paper
6. Thursday 12/10: Final Project Presentation

References:

[1] Health Care and Life Sciences Interest Group: http://esw.w3.org/topic/HCLSIG
[2] Linked Open Data Cloud: http://www4.wiwiss.fu-berlin.de/bizer/pub/lod-datasets_2009-03-05.html
[3] PubMed: http://pubmed.bio2rdf.org/sparql
[4] DailyMed has information about marketed drugs published by the National Library of Medicine: http://dailymed.nlm.nih.gov/dailymed/about.cfm, http://www4.wiwiss.fu-berlin.de/dailymed/
[5] LinkedCT has linked data sources of clinical trials: http://linkedct.org
[6] Drugbank has chemical, pharmacological and pharmaceutical properties of drug data with comprehensive drug target information: http://www.drugbank.ca, http://www4.wiwiss.fu-berlin.de/drugbank
[7] Diseasome has characteristics of disorders and disease genes linked by known disorder–gene associations: http://www4.wiwiss.fu-berlin.de/diseasome
[8] GO has data expressed using the Gene Ontology: http://www.geneontology.org , http://go.bio2rdf.org/sparql
[9] Exhibit: http://simile.mit.edu/wiki/Exhibit