ResearchIQ: Design of a Semantically Anchored Integrative Query Tool

Omkar Lele, MS, MBA, Satyajeet Raje, MS,
Po-Yin Yen, RN, PhD, Philip Payne, PhD, FACMI
Department of Biomedical Informatics, The Ohio State University, Columbus, OH

Abstract

An important factor influencing the pace of research activity is the ability of researchers to discover and leverage heterogeneous resources. Usually, researcher profiles, laboratory equipment, data samples, clinical trials, and other research resources are stored in heterogeneous datasets in large organizations. Emergent semantic web technologies provide novel approaches to discover, annotate and consequently link such resources. In this manuscript, we describe the design of Research Integrative Query (ResearchIQ) tool, a semantically anchored resource discovery platform that facilitates semantic discovery of local and publically available data through a single web portal designed for researchers in the biomedical informatics domain within The Ohio State University.

Introduction

There has been an explosion in the sheer extent and expanse of data and resources in recent years. Though this should translate to accelerated research, in reality the effect is insignificant. One major hindrance to the pace of research is the lack of support for researchers to discover and leverage heterogeneous resources [1]. This is often due to the information about data samples, clinical trials, genomics data as well as the researcher profiles stored in heterogeneous datasets, making it hard to find. No tools were available to simplify access to this data and information about resources stifles research activity and collaboration even in a single organization. The problem is even more pronounced when the information and data is spread across organizations.

Semantic technologies have been used successfully to facilitate research resource discovery [2,3]. For instance, PubMed [4], a widely adopted literature search portal in the biomedical community, relies on manually curated MeSH annotations. Vivo [5] finds research collaborators within and across organizations using expertise profiles. Eagle-I [6] uses semantic annotations to connect resources and leverages these connections to improve result relevance. Both Eagle-I and Vivo utilize semi-automated methods for generating semantic annotations. Most of the tools mentioned above require significant effort to manually assign or reassign semantic annotations for resource information. The weakness limits the number of resources as well as their heterogeneity [4,5] that can be represented within a system.

We present the Research Integrated Query (ResearchIQ) tool [7,8], an ontology-based semantic search portal, which aims to simplify access to heterogeneous datasets as well as to efficiently assist discovery of the discovery of biomedical and life science research resources at the Ohio State University.

The ResearchIQ System

ResearchIQ integrates heterogeneous biomedical and life sciences resources such as publications, clinical trials, expertise profiles, research grants, campus core laboratories, equipment and other software resources crucial for biomedical research. Its consolidated search capability allows researchers to find all the relevant research resources using a single unified search portal. The design of the user interface is driven by HCI design principles with particular focus on making it simple and easy to use.

ResearchIQ leverages ontology-based semantic annotations to link heterogeneous research resources. The semantic annotation process is fully automated, eliminating the need for human intervention to maintain resources, including updates to resources, within the ResearchIQ system. Consequently, ResearchIQ can be scaled easily to represent any large number of heterogeneous research resources. The semantic annotations are augmented with syntactic representations. The unique combination (semantic and syntactic) enhances result relevance significantly. ResearchIQ has a robust architecture that includes a Hadoop [9] based backend infrastructure. The architecture also provides APIs for other systems to interface with ResearchIQ. In fact, ResearchIQ is currently being used to enhance resource discovery within a research project management tool (CoRR – Computerized Research Record [10]) at The Ohio State University. Figure 1 shows the various components of the ResearchIQ system.

The system can be broken down into three core functionalities, the Annotation Pipeline**,** the Query Pipeline, and the Caching Pipeline. The annotation pipeline gets input from several data sources and compiles semantic and syntactic annotations into a RDF triple store and a Lucene [11] index respectively. The query pipeline captures end-user search criteria and translates it to query combinations for the semantic triple store and the syntactic index. The

caching pipeline caches results to improve the overall user search experience. In the following sections we discuss the design and implementation of ResearchIQ in greater detail.

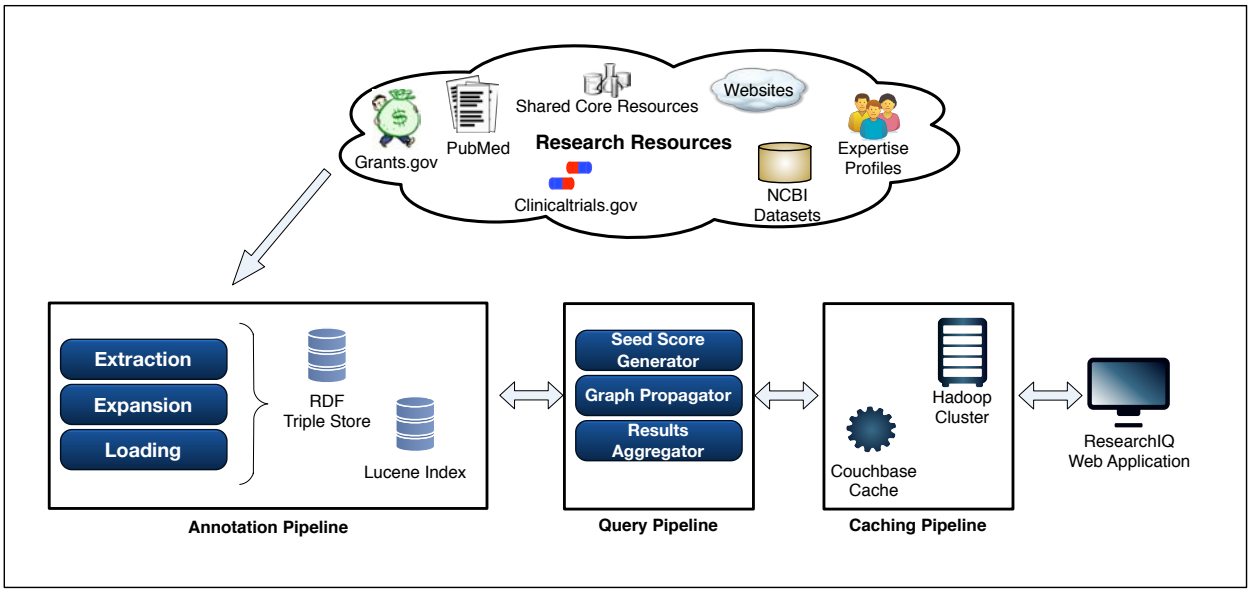

Figure 1. ResearchIQ System Components

Annotation Pipeline

As mentioned previously, the annotation pipeline aggregates heterogeneous research resources to generate the appropriate syntactic and semantic meta-information. Overall the pipeline performs three primary tasks viz. extraction, expansion and loading.

a) Extraction Phase: In this phase, every resource in ResearchIQ is classified as at least one of the resource types identified by Biomedical Resource Ontology (BRO) [12]. This is used for organizing the resources when disseminating results through the user interface. The MetaMap [13] tool is used to extract relevant (Unified Medical Language System) UMLS [14] concepts corresponding to the data and the meta-data associated with a resource. The resulting UMLS concepts are then used to annotate the resource within ResearchIQ. The modular architecture of ResearchIQ allows it to not be tied to tools like MetaMap, which can be easily replaced as required. ResearchIQ borrows data and object property definitions from several standard ontologies such as SKOS [15], NAO [16], FOAF [17] and BioSiteMaps [18] to form a comprehensive ontological representation capable of capturing all the required meta-information for the resources. We use the "skos:related" property to connect related resources, for example, an expert to his publications or a Principal Investigator to his or her clinical trial studies. The "nao:related" property is used to relate all the annotated UMLS concepts with their corresponding resources. The extraction phase also generates syntactic representations corresponding to the research resource data and meta-data. The syntactic representations include the actual ontological entities as well as syntactic signatures extracted from the resources directly, detailed in the loading phase. The output of the extraction phase is, thus, a set of syntactic and semantic representations of different types of resources with their respective meta-information and found UMLS concepts stored in an in-memory ontological graph format (as RDF [19] triples).

b) Expansion Phase: The expansion phase elicits details of the annotated concepts from UMLS and expands them with other concepts related to the annotated concepts in the UMLS database. These concepts along with their specific relationships to each other are added to the output from the extraction phase. The depth of this expansion of concepts is currently set at three as we observed significant overlap of concepts corresponding to expansion phases of various concepts and resources. However, the depth can also be controlled programmatically.

c) Loading Phase: The loading phase converts the in-memory ontological entities created in the extraction and expansion phase to distinct entries in the RDF triple store. We currently use Sesame [20] as the default RDF triple store, however, as mentioned previously, the modular architecture allows use of any other semantic store. In addition to semantics, the loading phase also loads syntactic representation of the resources into Apache Lucene indices. Note that the concepts along with the text associated with a resource are being indexed here. This is a critical distinction that gives the ResearchIQ query its semantic nature rather than just syntactic. The output of the annotation pipeline is a large number of semantic and syntactic representations corresponding to the input resources.

These representations are subsequently used to reason over the input search, as will be described in the Query Pipeline section, to generate semantically relevant search results.

Query Pipeline

The Query pipeline is executed as a result of an end-user specified search criteria. ResearchIQ allows the end-users to search via 1) UMLS concepts or 2) resource names. The query pipeline utilizes the underlying ontological structure generated by the annotation pipeline for querying. From a knowledge graph perspective, the search considers the UMLS concepts and the resources to be the graph nodes and the relationships to be the graph edges. Each search query triggers a graph-based propagation using the linkages (or the graph edges) to retrieve relevant results. As traversing the semantic knowledge graph generates results, the reasoning and resource discovery are arguably inherently semantic in nature. The query processor allows the user to form complex queries from individual concepts and resources as well. Additionally, the query portal allows end-users to utilize boolean operators to form complex search criteria. Thus, query logic implemented in ResearchIQ yields a set of highly precise semantically relevant resources, whether the search is performed using concepts or by resources. Figure 2 shows the high level algorithm used by the query pipeline.

a) Searching by UMLS concepts: In this case the user searches for resources relating to specific UMLS concepts. ResearchIQ uses these to generate a set of seed result resources from the Lucene index. Seed resources and corresponding score is obtained separately by querying for each concept term and its synonymous concept terms in the Lucene index. This generates a list of seed resources conceptually closest to the query. Next, for every searched concept, we find its related concepts from the UMLS ontology. Resources related to these concepts are added to the results however, with decayed scores. The further the resources are from the original set of UMLS concepts, the lower the score for the obtained resources. This process is executed recursively to obtain additional resources that are semantically connected to the original search. The score propagation is also modular and can be varied for different semantic relations between the concepts. The concept path through which a particular result was obtained is stored as part of the results. This allows us to provide provenance information to the user about why a particular resource is in the results of the query.

b) Searching by resource names: Alternatively, the user can search for resources directly by specifying resource names. However, the search results are not restricted to just the resource specified, but are extended to include other semantically relevant resources. As far as the algorithm for resource search is concerned, the directly specified resource gets the top score. The query engine then searches the RDF store for all the resources that are related to the specified resource using "skos:related" property. The UMLS concepts related to a particular resource are also used to find additional semantically related resources. This process follows the same technique for score calculation as was described in the previous section.

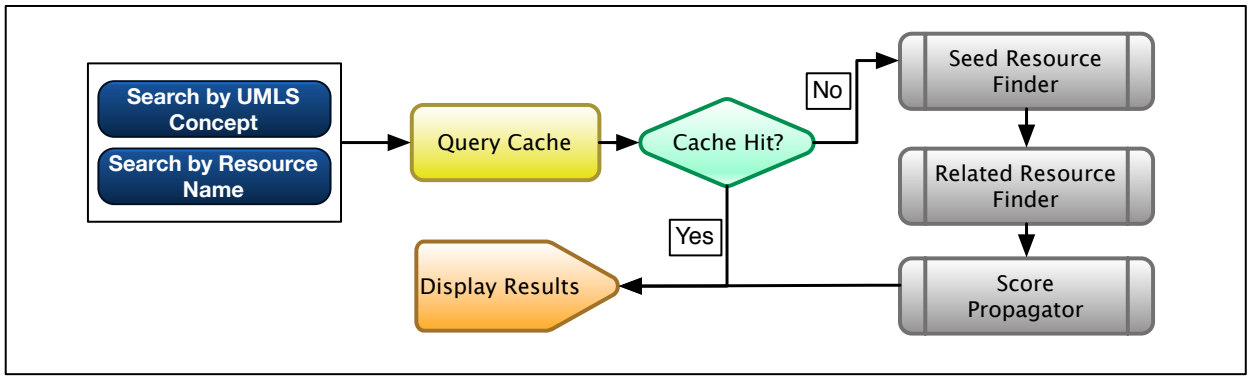

Figure 2. Query Pipeline Algorithm

Caching Pipeline

Due to the sheer size of the data and the number of resources being processed, ResearchIQ combines distributed computing with a highly scalable caching architecture to reduce the query response time, a very important factor of performance for any discovery engine [21]. It leverages the distributed processing capability of Hadoop to automate and accelerate the otherwise time-consuming data provisioning process. The idea of the cache mechanism is to pre-compute the results corresponding to search concepts and cache them beforehand. The cache is a key value store where the search term is the key and list of resources (result set) is the value. An automated caching pipeline is triggered every time the annotation pipeline is executed.

User Interface

The need for design grounded in HCI is critical for applications presenting large amounts of data [22]. As part of ResearchIQ development, we implemented a robust and intuitive user interface based on a preliminary needs assessment study [23] that was targeted to learn the information needs of prospective ResearchIQ users. It allows users to visualize and explore large number of search results efficiently as a hierarchical structure based on their semantic type. The analysis revealed that most of the user queries could be categorized into six categories and can be mapped to the high-level ontology nodes within the Biomedical Resource Ontology. We further classified the search results into seven resources types (Figure 3). We also adopted an event driven design for the user interface keeping the client (web-browser) lightweight that translated into lower waiting times for the users. Finally, we provide a hybrid search slider that is, to the best of our knowledge, unique to ResearchIQ among ontology based search systems. The slider allows users to filter results to only show syntactic, semantic or any mixture therein.

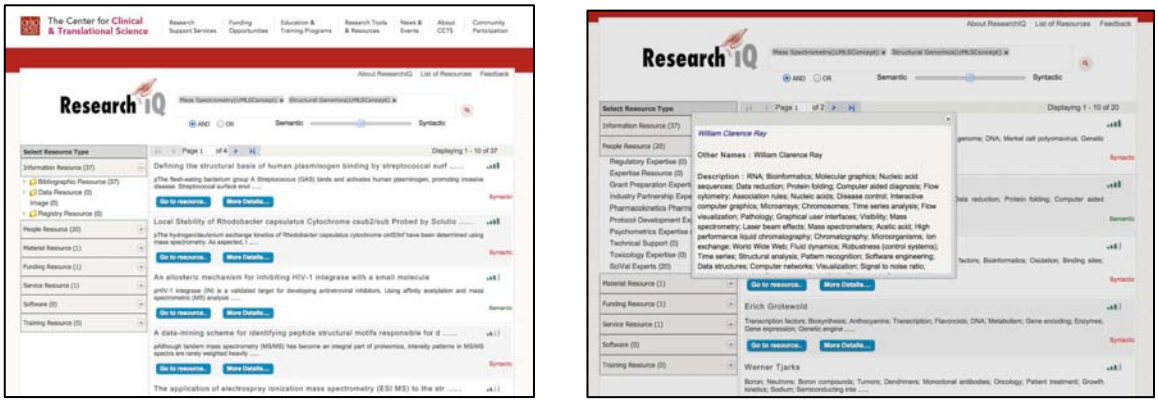

Figure 3. ResearchIQ User Interface.

System Improvement

To improve the system, we conducted a pilot usability evaluation to seek feedback from the biomedical research community. Eight researchers were recruited using purposive sampling, at The Ohio State University. A think aloud protocol was used to observe and understand users' interaction with ResearchIQ as well as their feedback. The study revealed that ResearchIQ enhanced search efficiency. However, researchers also experienced quite a steep learning curve trying to understand "semantics" and "semantically related resources". We see this as an opportunity to better design the user interface so as to keep the search simple and abstract out the technical semantic implementation. Further, it was observed that a purely semantic search gave a high degree of precision but low recall. Meaning that though the resources returned as search results were relevant, many relevant resources were not included. To improve recall performance we introduced the hybrid search mechanism to mix the results of the purely semantic search with traditional syntactic search.

The pilot study also revealed that the search response time was long (refer to Table 1 – no cache). Algorithmic analysis showed that this was due to latencies of complex SPARQL [24] queries in the semantic triple store. We found that without caching, the query time escalated drastically in direct relation to the number of search results returned. To mitigate this issue, parallel processing of results with a Hadoop backend feeding multilevel caches was introduced. This approach significantly reduced search latencies (Table 1).

Search Term	Number of Results	Latency (seconds)	
		No Cache	With Cache
"Diabetes Mellitus" (UMLS Concept)	381	109.19	< 1
"Cancer" (UMLS Concept)	6527	172.41	< 1

Table 1. Latency in seconds (for selected search terms) with and without caching

Discussion

ResearchIQ currently has about 75000 linked resources represented by about 1.5 million triples. This includes publications (e.g., PubMed abstracts), expertise (e.g., OSU:Pro faculty profiles), open federal grant opportunities (e.g., grants.gov), local shared resources and laboratory equipment (e.g., microarray analysis, microscopes), and open clinical studies (e.g., clinicaltrials.gov, StudySearch [25]). Since ResearchIQ is able to discover both

semantically and syntactically annotated resources, it gives users a unique unified platform to search across a broad range of distributed data sources. We describe the three core functionalities, the user interface, and summarize the system improvement based on a pilot usability evaluation study. Future work includes continued addition of data sources to the system along with further comprehensive usability evaluation and performance analysis.

ResearchIQ can be accessed via the web at http://researchiq.bmi.osumc.edu.

Support Acknowledgement

This work was supported by NIH/NCRR UL1-RR025755.

References

1. Payne P, Embi P. Clinical research informatics: challenges, opportunities and definition for an emerging domain. JAMIA. 2009; 16(3): p. 316-327.
2. Müller H, Kenny E, Sternberg P. Textpresso: an ontology-based information retrieval and extraction system for biological literature. PLoS Bio. 2004; 2(11): p. e309.
3. Bodenreider O. Biomedical ontologies in action: role in knowledge management, data integration and decision support. Yearbook of Me. Info. 2008; p. 67-79.
4. McEntyre J, Lipman D. PubMed: bridging the information gap. CMAJ. 2001; 164(9): p. 1317-1319.
5. Krafft D, Cappadona N, Caruso B, Corson-Rikert J, Devare M, Lowe B. Vivo: enabling national networking of scientists. Proc. of Web Sci. Conf. 2010; p. 1310-1313.
6. Segerdell E, Wilson M, Bashor T, Bourges-Waldegg D, Corday K, Frost R, et al. eagle-i: An Ontology-Driven Framework For Biomedical Resource Curation And Discovery. Nature Precedings. 2010.
7. Raje S. ResearchIQ: an end-to-end semantic knowledge platform for resource discovery in biomedical research. MS Thesis. The Ohio State University; 2012.
8. Borlawsky T, Lele O, Payne P. Research-IQ: Development and evaluation of an ontology-anchored integrative query tool. JBI. 2011; 44: p. S56-S62.
9. Apache Hadoop. [Online]. Available from: http://hadoop.apache.org/
10. Computerized Research Record (CoRR). [Online]; 2014. Available from: https://researchrecord.osu.edu
11. Apache Lucene. [Online]. Available from: http://lucene.apache.org/core
12. Tenenbaum J, Whetzel P, Anderson K, Borromeo C, Dinov I, Gabriel D, et al. The Biomedical Resource Ontology (BRO) to enable resource discovery in clinical and translational research. JBI.2011; 44(1): p.137-145.
13. Aronson A. Effective mapping of biomedical text to the UMLS metathesaurus: the MetaMap program. Proc. AMIA Symp. 2001; p.17-21.
14. Lindberg D, Humphreys B, McCray A. The Unified Medical Language System. Methods Inf Med. 1993; 32(4): p. 281-291.
15. Miles A, Bechhofer S. SKOS simple knowledge organization system reference. W3C Rec. 2009.
16. Scerri S, Sintek M, Elst L, Handschuh S. NEPOMUK Annotation Ontology Specification. OSCAF Recommendation. 2007.
17. Brickley D, Miller L. FOAF vocabulary specification. 2014.
18. Biositemaps White Paper. [Online]; 2008. Available from: http://biositemaps.ncbcs.org/Biositemaps_white_paper_v4.1.pdf
19. Resource Description Framework (RDF). [Online]. Available from: http://www.w3.org/RDF
20. OpenRDF: Sesame. [Online]. Available from: http://openrdf.callimachus.net
21. Chatra Raveesh S. Using the Architectural Tradeoff Analysis Method to Evaluate the Software Architecture of a Semantic Search Engine: A Case Study. MS Thesis. The Ohio State University; 2013.
22. Myers B. A Brief History of Human Computer Interaction Technology. ACM Interactions. 1998; 5(2): p. 44-54.
23. Yen P, Lele O, Borlawsky T, Payne P. ResearchIQ (Research Integrative Query) Preliminary Needs Assessment. In AMIA Summit on Clinical Research Informatics; 2012.
24. http://www.w3.org/TR/rdf-sparql-query/. [Online]; 2008.
25. Carpenter D, Kelsey B, Rice R, Reider C, Borlawsky T. StudySearch: A Tool for Connecting Potential Participants with Locally Recruiting Studies. AMIA Summits Transl Sci Proc. 2011.