A - Posteriori Integration for Life Sciences Data

Ali Hasnain[1]

Insight Center for Data Analytics, National University of Ireland, Galway
firstname.lastname@insight-centre.org

Abstract. Multiple datasets that add high value to biomedical research have been exposed on the web as part of the Life Sciences Linked Open Data (LS-LOD) Cloud. The ability to easily navigate through these datasets is crucial in order to draw meaningful biological co relations. However, navigating these multiple datasets is not trivial as most of these are only available as isolated SPARQL endpoints with very little vocabulary reuse. We propose an approach for Autonomous Resource Discovery and Indexing (ARDI), a set of configurable rules which can be used to discover links between biological entities in the LS-LOD cloud. We have catalogued and linked concepts and properties from 137 public SPARQL endpoints. The ARDI is used to dynamically assemble queries retrieving data from multiple SPARQL endpoints simultaneously.

1 Introduction

The advent of the World Wide Web [6] has enabled public publishing and consumption of information on a unique scale in terms of cost, accessibility and size. In the past few years, the linked open data cloud has earned a fair amount of attention and it is becoming the standard for publishing data on the Web [25].

One of the ambitions behind the linked data effort is the ability to create a Web of interlinked data which can be queried using a unified query language and protocol, regardless of where the data is stored. Core to this achievement is the adoption of the resource description framework (RDF) as the knowledge representation formalism as well as SPARQL protocol.

The life sciences domain has been the early adopters of linked data and, considerable portion of the Linked Open Data cloud is comprised of datasets from Life Sciences. The significant contributors includes the Bio2RDF project[1], Linked Life Data[2] and the W3C HCLSIG Linking Open Drug Data (LODD) effort[3]. Although the publication of datasets as RDF is a necessary step towards achieving unified querying of biological datasets, it is not enough to achieve the interoperability necessary to enable a query-able Web of life sciences data since it solves only the *"syntactic interoperability"* problem without addressing the *"semantic interoperability"* problem [5]. To achieve the ability for assembling queries encompassing multiple graphs hosted at various places, it is necessary

[1] http://bio2rdf.org/ (l.a.: 2016-03-31)
[2] http://linkedlifedata.com/ (l.a.: 2016-04-16)
[3] http://www.w3.org/wiki/HCLSIG/LODD (l.a.: 2016-05-16)

that vocabularies and ontologies are reused [21]. This can be achieved either by ensuring that the multiple datasets make use of the same vocabularies and ontologies known as *"a priori integration"* [8] or, using *"a posteriori integration"*, which makes use of mapping rules that change the topology of graphs such that integrated queries become possible. A *"posteriori"* solutions are favoured by Semantic Web technologies as these include mechanisms to describe two classes, for example describing experiments and said to be "the same" [8]. Our work focuses on a methodology to facilitate *"a posteriori integration"*.

2 Problem Statement

In the Life Sciences domain, Linked Data is extremely heterogeneous and dynamic [9]. This includes both syntactic as well as semantic heterogeneity. Also there is a recurrent need for *ad hoc* integration of novel experimental datasets due to the speed at which technologies for data capturing in this domain are evolving. As such, integrative solutions increasingly rely on federation of queries [24,10,1]. Standardisation of SPARQL 1.1, made now possible to assemble federated queries using the "SERVICE" keyword. To assemble queries encompassing multiple graphs distributed over different places, it is necessary that all datasets should be query-able using the same global schema [11,13]. This can be achieved either by ensuring that the multiple datasets make use of the same vocabularies and ontologies, an approach known as *"a priori integration"* or, using *"a posteriori integration"*, which makes use of mapping rules that change the topology of remote graphs to match the global schema [8] and the methodology to facilitate the latter approach is the focus of our research.

3 Relevancy

This problem seems important for researchers using Linked Open Data in general and Biomedical/ Bioinformatics researchers in specific.

4 Research question(s)

For LD to become a core technology in the LS domain, three issues need to be addressed, also provides baseline for research questions for our work: *i)* how to dynamically discover datasets containing data on biological entities (e.g. Proteins, Genes), *ii)* how to retrieve information about the same entities from multiple sources using different schemas, and *iii)* to identify, for a given query, the data with highest quality.

5 Hypothesis

Our hypothesis can be summarised as follows:
"Given heterogeneous data from a publicly available Life Sciences Linked Open Data corpus over distributed infrastructure, can we demonstrate improvements to SPARQL Query Federation for Knowledge Discovery by the generation of ARDI, an approach for indexing concepts and properties from distinct endpoints (partially) achieving a posteriori integration of data".

6 Approach

To address the aforementioned research questions, we introduce the notion of Autonomous Resource Discovery and Indexing (ARDI) – a representation of concepts and the links connecting these concepts. ARDI would not only help understand which data exists in each LS SPARQL endpoint, but more importantly enable assembly of multiple source-specific federated SPARQL queries. Since our work is based on data exposed as public SPARQL endpoints, it is important to analyze the content of each endpoint before creating ARDI. Hence our overall approach comprises of four distinct steps/ stages (Figure 1).

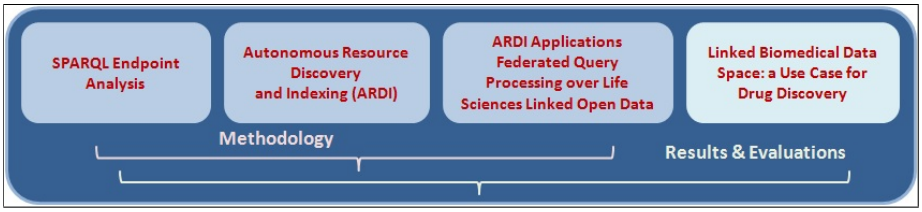

Fig. 1: Steps involved in our approach for addressing the problem

SPARQL Endpoint Analysis

The public SPARQL endpoints are planned to be analysed with two considerations *i. the content of a public SPARQL endpoint?* and *ii. how self descriptive these endpoints are?*. Analysing the content e.g. in terms of a) number classes, b) number of properties, c) list of classes, etc are necessary to investigate the size as well as finding similar data available at multiple datasources. Finding how much self descriptive any endpoint is important to know the structure of data stored at any endpoint in terms of class partitions, property partitions and well as nested partitions. With *self descriptive*, we mean the potential of any endpoint in order to express itself based on the data stored. In other words user can find the information regarding the endpoint and the data stored by simply querying the data itself. This includes the type of data (e.g. list of classes and properties), amount of data (e.g statistical snapshot regarding the entities, triples, classes and properties), structure of data (class partitions, property partitions and nested class/property partitions) and further classification of data (e.g. literals, blank nodes and IRIs). This analysis is presented by Hasnain et al [14], Such analysis provides a base line information regarding public SPARQL endpoint as we catalogue and link the content (ARDI) of these endpoints to support *"a posteriori" integration.*

Autonomous Resource Discovery and Indexing (ARDI)

The ARDI comprises a catalogue of LS-LOD and a set of functions to perform standard queries against it. The methodology for developing the ARDI consists

of two stages namely *catalogue generation* [11] and *link generation* [15]. The methodology for catalogue generation relies on retrieving all "types" (distinct concepts) from each SPARQL endpoint and all associated properties with corresponding instances. Data was retrieved from more than 130 public SPARQL endpoints[4], where the list was captured from publicly available Bio2RDF data sets and by searching for data sets in Datahub[5] tagged *"life science"* or *"healthcare"*. Hasnain et al, presented the methodology for catalogue generation [15] and link generation [11] using naïve, named entity and domain matching approaches for weaving the "types" together for set of query elements (Qe).

Query Engine

As the practical application of ARDI, a Domain Specific Query Engine is in a design phase that would offers a single-point-of-access for distributed life science data from reliable sources without extensive expertise in SPARQL query formulation. The ARDI identifies relevant triple patterns and matches types according to their labels as a basic semantic normalisation approach. New public endpoints are added through a cataloguing mechanism defined by ARDI. Query Engine would also provide provenance information covers *the sources queried*, *the number of triples returned* and *the retrieval time*.

Linked Biomedical Dataspace (LBDS)

The combination of different components and technologies ARDI (Cataloguing and Linking), Query Federation and Visual Query Explorer/ Aggregator) constitute a dataspace - we call it Linked Biomedical Data Space [12]. The Linked Biomedical Dataspace (LBDS) enables the semantically-enriched representation, exposure, interconnection, querying and browsing of biomedical data and knowledge in a standardised and homogenised way.

7 Related Work

Relevant areas for related Work are: i) Linked Data access methods, ii) Discovering SPARQL endpoints, iii) Cataloguing and Indexing, iv) Query Federation.

Linked Data access methods

There have been three methods provided to access content from knowledge bases published as Linked Data: DEREFERENCING, where IRIs of interest are looked up via HTTP; DUMPS, where the entire content of a dataset is made available for download; and SPARQL ENDPOINTS, where a query interface is provided over the local content. A more recent proposal – LINKED DATA FRAGMENTS [26] – has recently begun to gain attention. SPARQL endpoints push the burden from data consumers to producers: hosting such a public query service is expensive and as a

[4] `http://goo.gl/ZLbLzq`
[5] `https://datahub.io/` (l.a.: 2016-05-05)

result, endpoints may not be able to answer all queries for all consumer agents [7]. As an alternative to SPARQL endpoints, Verborgh et al. [26] propose methods for providing and organising multiple access methods to a Linked Dataset, including a lightweight "triple pattern fragment", which allows clients to request all triples matching a single pattern.

Discovering SPARQL endpoints

There are two high-level options for discovering SPARQL endpoints with relevant data: (1) flood the endpoints with queries, or (2) build a central search index. For example, federated SPARQL engines employ one or both of these strategies [22,24,2,1,4]. Paulheim et al. [20] looked at how to find a SPARQL endpoint containing content about a given Linked Data URI: using VoID descriptions and the DataHub catalogue. Buil-Aranda et al. [7] propose SPARQLES as a catalogue of SPARQL endpoints, but focus on performance and stability metrics rather than cataloguing content. Likewise, the analysis by Lorey [19] of public endpoints focused on characterising the performance offered by these services rather than on the problem of discovery.

Cataloguing and Linking

Ontology alignment approaches can not be used for cataloguing as these do not make use of domain rules (e.g. for two same sequences, qualifies for same gene) nor the use of URI pattern matching for alignment [11]. Approaches such as the VoID [3] and the SILK Framework [27] enable the identification of rules for link creation, but require extensive knowledge of the data prior to links creation. Our approach for link creation is a combination of the several linking approaches as already explained by Hasnain et. al [11]: *i)* similarly to ontology alignment, we make use of label matching to discover concepts in LOD that should be mapped to a set of *Qe*, *ii)* we create *"bags of words"* for discovery of schema-level links similar to the approach taken by BLOOMS, and *iii)* as in SILK, we create domain rules that enable the discovery of links.

SPARQL Federation Systems

Advances in federated query processing methods over the Web of Data have enabled the development of federated query engines (QE). Each of these QE have slightly different goals and thus make different compromises between speed, completeness, and flexibility. Quilitz et al. [23] proposed DARQ. It makes use of service descriptions for relevant data source selection. Langegger et al. [18] propose a solution using a mediator approach, which continuously monitors the SPARQL endpoints for any dataset changes for automatic updates. Schwarte et al. [24] propose FedX, an index-free query federation for the Web of Data. SPLENDID [10] makes use of Vocabulary of Interlinked Datasets (VoID) descriptions along with SPARQL ASK queries to select the list of relevant sources

for each triple pattern. Kaoudi et al. [17] propose a federated query technique on top of distributed hash tables (DHT) to minimise the query execution time and the bandwidth consumption. Acosta et al. [1] present ANAPSID, an adaptive query engine that adapts query execution schedulers to endpoints data availability and run-time conditions. Avalanche [4] gathers endpoint datasets statistics and bandwidth availability on-the-fly before the query federation.

8 Preliminary Results

Results for our ARDI approach has been published [11], [15]. We evaluated the performance of our catalogue generation methodology and recorded the times taken to probe instances through endpoint analysis of 12 endpoints whose underlying data sources were considered relevant for drug discovery. The cataloguing experiments were carried out on a standard machine with 1.60Ghz processor, 8GB RAM using a 10Mbps internet connection. Best fit regression models were then calculated (Fig. 2). It took less than 1000000 milliseconds (<16 minutes) to catalogue seven of the SPARQL endpoints, and a gradual rise with the increase in the number of available concepts and properties. We obtained two power regression models ($T = 29206 * C_n^{1.113}$ and $T = 7930 * P_n^{1.027}$) to help extrapolate time taken to catalogue any SPARQL endpoint with a fixed set of available concepts (C_n) and properties (P_n), with R^2 values of 0.641 and 0.547 respectively. Using these models and knowing the total number of available concepts/properties, a developer could determine the approximate time (ms) as a vector combination. KEGG and SGD endpoints took an abnormally large amount of time for

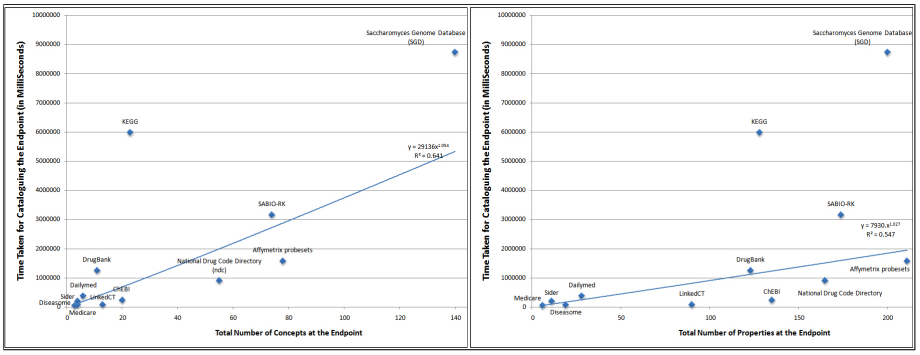

Fig. 2: Time taken to catalogue 12 SPARQL endpoints

cataloguing than the trendline. We also evaluated the performance of our Link Generation methodology by comparing it against the popular linking approaches. Using WordNet thesauri we attempted to automate the creation of bags of related words using 6 algorithms [11]: Jing & Conrath, Lin, Path, Resnik, Vector and WuPalmer with unsatisfactory results (Figure 3(c)). Our linking approaches resulted in better linking rate as shown in Figure 3(a,b)

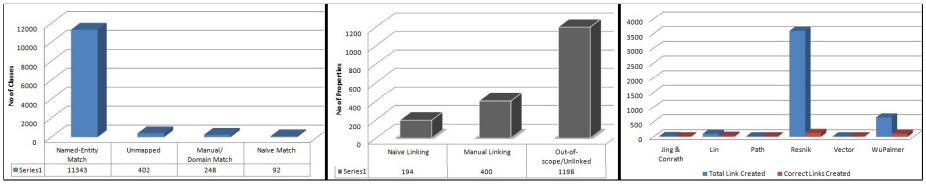

Fig. 3: (a) Number of Classes Linked, (b) Number of Properties Linked, (c) Number of Classes linked through available similarity linking approaches

9 Evaluation Plan

Our Evaluation plan will span over evaluating our approach in terms of i) SPARQL endpoint analysis, ii) ARDI (cataloguing and linking) and iii) Query Federation System. All these stages have different evaluation criteria.
SPARQL Endpoint Analysis:Criteria is twofold: (i) using VoID as a bar, to empirically investigate the extent to which public endpoints can describe their own content, and (ii) to build and analyse the capabilities of a best-effort online catalogue of current endpoints based on the (partial) results collected.
ARDI (cataloguing and linking): Cataloguing and Linking results along with the evaluation in terms of i) time taken, ii) number of concepts and properties catalogued, and iii) correct vs incorrect links has been published[11], [15].
Query Federation System: For evaluation the query federation system, we define source selection efficiency in terms of (a) total number of triple-wise sources selected (#TP), (b) SPARQL ASK requests used (#AR; to obtain (a)), and (c) the source selection time (SST). Based on this criteria we aim to evaluate our system with FedX a state of the art query engine using a test bed of ten real time datasets with 20 real time queries (a publication under review).

10 Reflections

Focusing on the problem of finding relevant SPARQL endpoints and analysing, we may miss relevant Linked Datasets that do not offer a SPARQL endpoint. According to statistics by Jentzsch et al. [16], only 68% of the Linked Datasets surveyed provided a SPARQL endpoint. However, our focus is specifically on the problem of relevant SPARQL endpoints, which we argue is a sufficiently noteworthy problem in and of itself. Current experiments and evaluation uses a set of *Qe*, which were defined in a context of drug discovery. The number of classes per endpoint varied from a single class to a few thousands. Our initial exploration of the LSLOD revealed that only 15% of classes are reused. However, this was not the case for properties, of which 48.5% are reused. Multiple challenges faced which can hinder the applicability of our approach:

– Some endpoints return timeout errors when a simple query (`SELECT DISTINCT ?Concept WHERE {[ ] a ?Concept}`) is issued.

– Some endpoints have high downtime and cannot be generally relied.
– Many endpoints provide non-deferenceable URI and some derefenceable URI do not provide a "type" for the instance.

Acknowledgement

This research has been supported in part by Science Foundation Ireland under Grant Number SFI/12/RC/2289. The author would also like to acknowledge Dietrich Rebholz-Schuhmann being PhD supervisors.

References

1. Acosta, M., Vidal, M., Lampo, T., Castillo, J., Ruckhaus, E.: ANAPSID: an adaptive query processing engine for SPARQL endpoints. In: International Semantic Web Conference (ISWC). pp. 18–34. Springer (2011), http://dx.doi.org/10.1007/978-3-642-25073-6_2
2. Akar, Z., Halaç, T.G., Ekinci, E.E., Dikenelli, O.: Querying the Web of Interlinked Datasets using VOID Descriptions. In: Linked Data On the Web (LDOW). CEUR (2012)
3. Alexander, K., Hausenblas, M.: Describing linked datasets-on the design and usage of void, the'vocabulary of interlinked datasets. In: In Linked Data on the Web Workshop (LDOW 09), in conjunction with WWW09. Citeseer (2009)
4. Basca, C., Bernstein, A.: Querying a messy web of data with Avalanche. J. Web Sem. 26, 1–28 (2014)
5. Bechhofer, S., Buchan, I., De Roure, D., Missier, P., et al.: Why linked data is not enough for scientists. Future Generation Computer Systems 29(2), 599–611 (2013)
6. Berners-Lee, T., Fischetti, M., Foreword By-Dertouzos, M.L.: Weaving the Web: The original design and ultimate destiny of the World Wide Web by its inventor. HarperInformation (2000)
7. Buil-Aranda, C., Hogan, A., Umbrich, J., Vandenbussche, P.Y.: Sparql web-querying infrastructure: Ready for action? In: The Semantic Web–ISWC 2013, pp. 277–293. Springer (2013)
8. Deus, H.F., Prud'hommeaux, E., Miller, M., Zhao, J., Malone, J., Adamusiak, T., et al.: Translating standards into practice–one semantic web API for gene expression. Journal of biomedical informatics 45(4), 782–794 (2012)
9. Goble, C., Stevens, R., Hull, D., et al.: Data curation+ process curation= data integration+ science. Briefings in bioinformatics 9(6), 506–517 (2008)
10. Görlitz, O., Staab, S.: Splendid: Sparql endpoint federation exploiting void descriptions. In: Proceedings of the 2nd International Workshop on Consuming Linked Data, Bonn, Germany (2011)
11. Hasnain, A., Fox, R., Decker, S., Deus, H.F.: Cataloguing and linking life sciences LOD Cloud. In: 1st International Workshop on Ontology Engineering in a Data-driven World collocated with EKAW12 (2012)
12. Hasnain, A., Kamdar, M.R., Hasapis, P., Zeginis, D., Warren Jr, C.N., et al.: Linked Biomedical Dataspace: Lessons Learned integrating Data for Drug Discovery. In: International Semantic Web Conference (In-Use Track), October 2014 (2014)
13. Hasnain, A., Mehmood, Q., e Zainab, S.S., Decker, S.: A provenance assisted roadmap for life sciences linked open data cloud. In: Knowledge Engineering and Semantic Web, pp. 72–86. Springer (2015)

14. Hasnain, A., Mehmood, Q., e Zainab, S.S., Hogan, A.: Sportal: Profiling the content of public sparql endpoints. International Journal on Semantic Web and Information Systems (IJSWIS) 12(3), 134–163 (2016), http://www.igi-global.com/article/sportal/160175
15. Hasnain, A., e Zainab, S.S., Kamdar, M.R., Mehmood, Q., Warren Jr, C.N., Fatimah, Q.A., Deus, H.F., Mehdi, M., Decker, S.: A roadmap for navigating the life sciences linked open data cloud. In: Semantic Technology, pp. 97–112. Springer (2014)
16. Jentzsch, A., Cyganiak, R., Bizer, C.: State of the lod cloud. Online Report (September 2011), http://lod-cloud.net/state/
17. Kaoudi, Z., Kyzirakos, K., Koubarakis, M.: Sparql query optimization on top of dhts. In: Proceedings of the 9th international semantic web conference on The semantic web - Volume Part I. pp. 418–435. ISWC'10 (2010)
18. Langegger, A., Wöß, W., Blöchl, M.: A semantic web middleware for virtual data integration on the web. In: Proceedings of the 5th European semantic web conference on The semantic web: research and applications. pp. 493–507. ESWC'08 (2008)
19. Lorey, J.: Identifying and determining SPARQL endpoint characteristics. IJWIS 10(3), 226–244 (2014), http://dx.doi.org/10.1108/IJWIS-03-2014-0007
20. Paulheim, H., Hertling, S.: Discoverability of SPARQL Endpoints in Linked Open Data. In: International Semantic Web Conference (ISWC) Posters & Demos. pp. 245–248. Springer (2013)
21. Polleres, A.: Semantic web technologies: From theory to standards. In: 21st National Conference on Artificial Intelligence and Cognitive Science, NUI Galway (2010)
22. Quilitz, B., Leser, U.: Querying distributed RDF data sources with SPARQL. In: European Semantic Web Conference (ESWC). pp. 524–538. Springer (2008)
23. Quilitz, B., Leser, U.: Querying distributed rdf data sources with sparql. In: Proceedings of the 5th European semantic web conference on The semantic web: research and applications. pp. 524–538. ESWC'08 (2008)
24. Schwarte, A., Haase, P., Hose, K., Schenkel, R., Schmidt, M.: FedX: A federation layer for distributed query processing on Linked Open Data. In: Extended Semantic Web Conference (ESWC). pp. 481–486. Springer (2011)
25. Smith, B., Ashburner, M., Rosse, C., Bard, J., Bug, W., Ceusters, W., Goldberg, L.J., Eilbeck, K., Ireland, A., Mungall, C.J., et al.: The obo foundry: coordinated evolution of ontologies to support biomedical data integration. Nature biotechnology 25(11), 1251–1255 (2007)
26. Verborgh, R., Hartig, O., Meester, B.D., Haesendonck, G., Vocht, L.D., Sande, M.V., Cyganiak, R., Colpaert, P., Mannens, E., de Walle, R.V.: Querying datasets on the Web with high availability. In: International Semantic Web Conference (ISWC). pp. 180–196. Springer (2014), http://dx.doi.org/10.1007/978-3-319-11964-9_12
27. Volz, J., Bizer, C., Gaedke, M., Kobilarov, G.: Discovering and maintaining links on the web of data. Springer (2009)