Utilizing Domain-Specific Keywords for Discovering Public SPARQL Endpoints: A Life-Sciences Use-Case

Muntazir Mehdi, Aftab Iqbal, Ali Hasnain,
Yasar Khan, Stefan Decker, and Ratnesh Sahay
INSIGHT Centre for Data Analytics, National University of Ireland, Galway
{firstname.lastname}@deri.org

ABSTRACT

The LOD cloud comprises of billions of facts covering hundreds of datasets. In accordance with the Linked Data principles, these datasets are connected by a variety of typed links, forming an interlinked "Web of Data". The growing diversity of the Web of Data makes it more and more challenging for publishers to find relevant datasets that could be linked to, particularly in specialist domain-specific settings. This paper thus proposes a baseline method to automatically identify a list of public SPARQL endpoints whose content is deemed relevant to a local dataset based on queries generated from a local set of domain-specific keywords.

Keywords

Linked Open Data (LOD) Cloud, Web of Data, SPARQL, Healthcare and Life Sciences

1. INTRODUCTION

The Linking Open Data (LOD) Cloud[1] lists 295 Linked Datasets, which, according to publisher statistics, incorporate over 30 billion facts. With regards to accessing this content, aside from crawling the raw data, 68% of the LOD datasets offer a link to at least one SPARQL endpoint that can be used to directly query the dataset. The Datahub catalogue[2] lists more than 427 such SPARQL endpoints [2].

The LOD Cloud also claims that there are over 500 million links across datasets. From the perspective of a consumer, these links allow for recursively discovering and navigating detailed information about related entities elsewhere on the Web. From the perspective of a publisher, links encourage modularity, where high-quality links (once in place) can reduce the amount of content they need to host. From the perspective of the Web, these links form the mesh upon which the Web of Data is based.

[1]`http://lod-cloud.net/state/`; l.a. 2013/12/06
[2]`http://datahub.io/group/lodcloud`; l.a. 2013/12/06

Table 1: Example terms for Clinical Partners (CP)

	Domain	Example Terms
CHUV	Cardiovascular	Coronary Heart Disease
	Psychiatric Disorder	Major Depressive disorder
	Migraine	Migraine cumulative (with aura)
CING	Diabetes	Urine Microalbumin
	Breast Cancer	Breastfeeding duration
	Neurology	Spinal Muscular Atrophy
ZEINCRO	Concomitant Meds	Hepatic or Biliary
	Respiratory	Rate of Spirometry
	Medical History	Musculoskeletal

But creating links is a challenging task for publishers. Addressing this challenge, a number of linking frameworks, such as Silk [12] and LIMES [10], have been proposed to help publishers link their local datasets to a remote LOD dataset through a specified SPARQL endpoint. However, given that there are now hundreds of public SPARQL endpoints, and many of these endpoints are black-boxes that do not describe their content [2], a still-more fundamental question has not been tackled: *how can publishers find SPARQL endpoints that are relevant targets for links in the first place*?

In this paper, we investigate a method that formulates a set of exact-literal candidates from a domain-specific keyword phrase, generating common variations based on stopword removal, word permutations, etc. To determine the relevance of (public) SPARQL endpoints, we probe them with queries looking for these literals. This method does not require expensive REGEX filters or non-standard full-text-search.

The rest of the paper is structured as follows: Section 2 presents a methodology that identifies potentially relevant data sources. Section 3 presents evaluation of our method for three clinical terminologies in our use-case, seeking relevant LOD datasets for linking. Section 4 discusses related work. But first we introduce our motivating scenario, where our work is inspired by a project with three clinical organisations that wish to publish their corpora as Linked Data.

Motivating Scenario. The Linked2Safety[3] consortium (an EU Project) includes three clinical partners namely, Univer-

[3]`http://www.linked2safety-project.eu/`

Table 2: Distribution of $KwList$ cardinalities

	$\|KwList\| = 1$	$\|KwList\| = 2$	$\|KwList\| = 3$
CHUV	140	70	5
CING	92	58	24
ZEINCRO	138	7	5

sity Hospital Lausanne (CHUV)[4], Cyprus Institute of Neurology and Genetics (CING)[5], and ZEINCRO[6]. One of the core goals of the project is to publish biomedical datasets provided by the clinical partners as high-quality Linked Data. Each clinical partner has provided clinical terminologies for their specialised domain with 150–215 terms each; Table 1 provides some examples. After RDF-ising the data, the next step is to identify potential LOD datasets that overlap with these domains of study and that are thus targets for linking.

The life-sciences community has been very active within the Linking Open Data movement: 41 datasets on the LOD cloud are classified as specialising in the "Life Sciences" domain and 70 SPARQL endpoints have been made available by these publishers, most prominently by the Bio2RDF[7] and Linked Life Data[8] initiatives. We propose methods that take as input aforementioned clinical terminologies and produce as output a list of potentially relevant SPARQL endpoints for linking. Specifically in this paper, we propose a method as a pre-processing step that generates a set of Query Terms or Keywords based on which a particular SPARQL endpoint is probed for relevance.

2. TERM EXTRACTION

We now present methods for creating exact query literals from domain-specific keyword phrases such as the clinical terms from our motivating scenario. These literals are then searched against public endpoints.

The algorithm, *Query Term Extractor* (QTermEx, Algorithm 1), takes as input a terminology, in this case a set of Clinical Terms (*CTerms*), and generates a set of *Query Terms* (*QTerms*).

All *CTerm*s in the terminology are iterated over. The input *CTerm* is first pre-processed by replacing junk characters with spaces (*JunkCharSet*; e.g., punctuation, parentheses, symbols, etc.), by removing stop words (*SWSet*, e.g., "the", "and', etc.) and by removing general terms (*GTSet*; e.g., "duration", "rate", "family", etc.). Junk characters are pre-defined. Stop words are collected from the terminology using the Ranks.nl text-analysis tool[9]. General terms are specific to a given terminology and are defined by domain experts; these general terms are used to reduce the number of *QTerms* created in the final output. Taking an example *CTerm* "Migraine cumulative (with aura)", first the parentheses will be removed as junk characters, "with" will be removed as a stop-word and "aura" will be removed as a general term (if defined). A list of unique token words, *KwList*, is computed as a result. Similarly for URIs, where we take an example *CTerm* `http://www.chuv.ch/variables/schizophrenia/code:SZAPU2`, first characters like (':', '/', '.') are replaced by spaces and words like ("http", "www", "chuv", "ch", "variables", "code") are eliminated as stop-words or as general terms (if defined).

Algorithm 1: QTermEx: Extracts Query Terms

```
Input: A finite set of Clinical Terms (CTerms)
Output: A finite set of Query Terms (QTerms)
$JunkCharSet$ := set of Junk Characters;
$SWSet$ := set of Stop Words;
$GTSet$ := set of General Terms;
$QTerms := \emptyset$;
for $CTerm$ in $CTerms$ do
    $CTerm'$ := replace each occurrence of $JunkCharSet$ in $CTerm$ with space;
    $TokenList$ := tokenise $CTerm'$ based on spaces;
    $KwList$ := empty list;
    for $Token$ in $TokenList$ do
        if $Token$ not in $SWSet, GTSet$ or $TokenList'$ then
            Add $Token$ to $KwList$;
    $QTerms'$ := all $n$-grams from $KwList$ that preserve ordering;
    $QTerms := QTerms' \cup QTerms$;
return $QTerms$;
```

All combinations of n-grams that preserve the ordering of the original input term (but potentially skip terms) are then computed from the resulting *KwList*. Thus, the result of our former example consist of three query terms: "Migraine", "cumulative" and "Migraine cumulative". These n-grams are added to the output *QTerms*. The total number of such n-gram query terms is $2^k - 1$ for $k = |KwList|$. For our use-case, Table 2 lists the distribution of cardinalities for $KwList$: the exponential combination of tokens into n-grams does not pose a significant problem since the size of $KwList$ never exceeds 3. Thus our algorithm is best suited to terminologies with concise domain-specific phrases (after the removal of stop words and general terms).

The *QTerms* generated from our algorithm QTermEx can be used to query the SPARQL endpoints. We call this a Direct Matching (DM) approach, where each *QTerm* is used to generate a single query literal. Each literal is used to generate a simple query as given in Listing 1, which can be used to probe for relevant endpoints.

```
SELECT ?s ?p
WHERE {
   ?s ?p "QTerm" .
}
```

Listing 1: SPARQL Query

3. EVALUATION

All experiments were conducted on a computer running 64-bit Windows 7 OS, with 4GB RAM and an Intel Core i3 (2.13 GHz) CPU.

Given three sets of raw clinical terms (*CTerms*) as input from three clinical partners, we generate three sets of query terms (*QTerms*) by applying the QTermEx algorithm. The number of *CTerms* with their corresponding

[4]`http://www.chuv.ch/`; l.a. 2013/12/06
[5]`http://www.cing.ac.cy/`; l.a. 2013/12/06
[6]`http:/www.zeincro.com/`; l.a. 2013/12/06
[7]`http://bio2rdf.org/`; l.a. 2013/12/06
[8]`http://linkedlifedata.com/`; l.a. 2013/12/06
[9]`http://www.ranks.nl/`; l.a. 2013/12/06

Table 3: Number of CTerms and QTerms

Dataset	CTerms	QTerms
`CHUV`	215	385
`CING`	174	434
`ZEINCRO`	150	194

generated *QTerms* is presented in Table 3, where it can be clearly seen that when compared with input *CTerms*, the number of output *QTerms* increases by a factor of 1.8× for `CHUV`, 2.5× for `CING` and 1.3× for `ZEINCRO`. The higher factor for `CING` is due to having a higher number of relevant keywords in each *CTerm* (which in turn affects performance).

We implemented the QTermEx algorithm using Java and recorded the execution time. For experimental reasons we executed the algorithm 10 times for each dataset and recorded runtimes for three different cases: 1) Best Case, 2) Average Case, and 3) Worst Case. The recorded values are presented in Figure 1, where it can be seen that although the number of input *CTerms* for the `CING` dataset are fewer than for the `CHUV` dataset and slightly higher then `ZEINCRO` dataset, the relative execution time for the `CING` dataset is much higher then `ZEINCRO` dataset and slightly lower than the `CHUV` dataset.

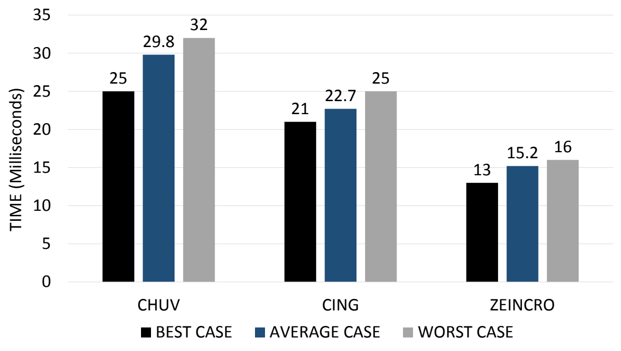

Figure 1: Execution Time

4. RELATED AND FUTURE WORK

A number of Linked Data search engines have been proposed in the literature, including FactForge [1], FalconS [4], Sindice [11], SWSE [7], etc. These engines allow for performing full-text search over local indexes of Linked Data. However, these systems focus on returning entities and/or documents as results, not SPARQL endpoints that are relevant for generating links.

A plethora of works have looked at creating links from local domain-specific datasets to the LOD cloud. These include (but are far from limited to) work by Hassanzadeh and Consens on generating links for movie-related data [6], work by Lebo et al. on generating links from governmental data [8], works by Llavori et al. [9], Hasnain et al. [5], and Callahan et al. [3] on linking biomedical data, and so forth. Again however, these works presuppose that the target datasets for linking are already known.

To the best of our knowledge, no work has presented methods to discover public SPARQL endpoints that are most relevant to a list of domain-specific keywords, nor has any work presented methods to automatically discover SPARQL endpoints that are potential targets for linking.

As an extension to current work, we are working on a SPARQL based searching scheme that will extensively use the extracted keywords from clinical terminologies. And based on the keywords, the scheme will recommend relevant publicly available SPARQL endpoints. The LOD datasets of the corresponding recommended SPARQL endpoints can be then used for interlinking.

5. ACKNOWLEDGEMENTS

This publication has emanated from research supported in part by a research grant from Science Foundation Ireland (SFI) under Grant Number SFI/12/RC/2289 and EU project Linked2Safety (contract Number 288328)

6. REFERENCES

[1] B. Bishop, A. Kiryakov, D. Ognyanov, I. Peikov, Z. Tashev, and R. Velkov. Factforge: a fast track to the web of data. *Semantic Web*, 2(2):157–166, Apr. 2011.

[2] C. Buil-Aranda, A. Hogan, J. Umbrich, and P.-Y. Vandenbussche. Sparql web-querying infrastructure: Ready for action? In *ISWC*, 2013.

[3] A. Callahan, J. Cruz-Toledo, P. Ansell, and M. Dumontier. Bio2RDF Release 2: Improved Coverage, Interoperability and Provenance of Life Science Linked Data. In *ESWC*, pages 200–212, 2013.

[4] G. Cheng and Y. Qu. Searching Linked Objects with Falcons: Approach, Implementation and Evaluation. *IJSWIS*, 5(3):49–70, 2009.

[5] A. Hasnain, R. Fox, S. Decker, and H. F. Deus. Cataloguing and linking life sciences lod cloud. In *OEDW*, 2012.

[6] O. Hassanzadeh and M. P. Consens. Linked movie data base. In *LDOW*, 2009.

[7] A. Hogan, A. Harth, J. Umbrich, S. Kinsella, A. Polleres, and S. Decker. Searching and browsing Linked Data with SWSE: the Semantic Web Search Engine. *J. Web Sem.*, 9(4):365–401, 2011.

[8] T. Lebo and G. T. Williams. Converting governmental datasets into linked data. In *Proceedings of the 6th International Conference on Semantic Systems*, page 38. ACM, 2010.

[9] R. B. Llavori, E. Jiménez-Ruiz, and V. Nebot. Exploring and linking biomedical resources through multidimensional semantic spaces. *BMC Bioinformatics*, 13(S-1):S6, 2012.

[10] A.-C. Ngonga Ngomo and S. Auer. Limes - a time-efficient approach for large-scale link discovery on the web of data. In *Proceedings of IJCAI*, 2011.

[11] E. Oren, R. Delbru, M. Catasta, R. Cyganiak, H. Stenzhorn, and G. Tummarello. Sindice.com: a document-oriented lookup index for open linked data. *IJMSO*, 3(1):37–52, 2008.

[12] J. Volz, C. Bizer, M. Gaedke, and G. Kobilarov. Silk - a link discovery framework for the web of data. In *LDOW*, 2009.