Including Co-referent URIs in a SPARQL Query

Christian Y A Brenninkmeijer[1], Carole Goble[1], Alasdair J G Gray[1], Paul Groth[2], Antonis Loizou[2], and Steve Pettifer[1]

[1] School of Computer Science, University of Manchester, UK.
[2] Department of Computer Science, VU University of Amsterdam, The Netherlands.

Abstract. Linked data relies on instance level links between potentially differing representations of concepts in multiple datasets. However, in large complex domains, such as pharmacology, the inter-relationship of data instances needs to consider the context (e.g. task, role) of the user and the assumptions they want to apply to the data. Such context is not taken into account in most linked data integration procedures. In this paper we argue that dataset links should be stored in a stand-off fashion, thus enabling different assumptions to be applied to the data links during query execution. We present the infrastructure developed for the Open PHACTS Discovery Platform to enable this and show through evaluation that the incurred performance cost is below the threshold of user perception.

1 Introduction

A key mechanism of linked data is the use of equality links between resources across different datasets. However, the semantics of such links are often not trivial: as Halpin et al. have shown sameAs, is not always sameAs [12]; and equality is often context- and even task-specific, especially in complex domains. For example, consider the drug *"Gleevec"*. A search over different chemical databases return records about two different chemical compounds – *"Imatinib"* and *"Imatinib Mesylate"* – which differ in chemical weight and other characteristics. When a user requires data to perform an analysis of the compound, they would require that these compounds are kept distinct. However, if they were investigating the interactions of *"Gleevec"* with targets (e.g. proteins) then they would like these records related. This requires two different sets of links between the records: one that links data instances based on their chemical structure and another where the data instances are linked based on their drug name. Users require the ability to switch between these alternative scientific assumptions [4].

To support this requirement, we have developed a new approach that applies context-dependent sets of data instance equality links at query time. The links are stored in a stand-off fashion so that they are not intermingled with the datasets, and are accessible through services such as BridgeDB [14] or sameas.org[3] [7]. This allows for multiple, context-dependent linksets that can

[3] `http://sameas.org/` accessed July 2013

evolve without impacting the underlying datasets and can be activated depending upon the user's context. This flexibility is in contrast to both linked data and traditional data integration approaches, that expose a single preconfigured view of the data. However, the use of stand-off mappings should not impact query evaluation times from the user's perspective.

In this paper, we present a query infrastructure to expand the instance URIs in a given query with equivalent URIs drawn from linksets stored in a stand-off fashion. It is assumed that such a query specifies which properties are to be taken from each of the data sources, i.e. a global-as-view query [8], and as such we do not consider ontology alignment issues [9]. This query infrastructure allows the linkset store to decide about which links are activated under a specific context; this will be explored in future work. We present a performance evaluation of the query infrastructure in comparison with a typical linked data approach (Section 4), which shows that query execution times are within the required performance characteristics (i.e. interactive speed) while allowing for contextual views. After the evaluation, we discuss related work and conclude.

2 Motivation and Context

This work is motivated by the needs of pharmacology researchers within the context of the Open PHACTS project[4]; a public-private partnership aimed at addressing the problem of public domain data integration for both academia and major pharmaceutical companies [18]. Data integration is a prerequisite for modern drug discovery as researchers need to make use of multiple information sources to find and validate potential drug candidates. The integration of knowledge from these disparate sources presents a significant problem to scientists, where intellectual and scientific challenges are often overshadowed by the need to repeatedly perform error-prone and tedious mechanical integration tasks.

A key challenge in the pharmacological domain is that the complexity of the domain makes developing a single view on the pharmacological data space difficult. For example, in some areas of the life sciences it is common to refer to a gene using the name of the protein it encodes (a geneticist would interpret *"Rhodopsin"* as meaning 'the gene RHO that encodes the protein Rhodopsin') whereas in other areas the terms are treated as referring to distinct, non interchangeable entities. Clearly *"RHO"* and *"Rhodopsin"* cannot in all cases be considered as synonyms, since they refer to different types of biological concept. Another challenge is that the domain scientists require different data record links depending on the context of their task. For example, when searching for information about *"Protein Kinase C Alpha"*, the scientist may want information returned for that protein as it exists in humans[5], mice[6] or both. Additionally, when connecting across databases one may want to treat these proteins as equal

[4] `http://www.openphacts.org/` accessed July 2013
[5] `http://www.uniprot.org/uniprot/P17252` accessed July 2013
[6] `http://www.uniprot.org/uniprot/P20444` accessed July 2013

in order to bring back all possible information whereas in other cases (e.g. when the researcher is focused on humans) only information on the particular protein should be retrieved. In both of the above cases, a single normalized view is impossible to produce.

To address the need for such multiple views as well as other data integration issues, the Open PHACTS Discovery Platform has been developed using semantic technologies to integrate the data. The overall architecture, and the design decisions for it, are detailed in [10]. Here, we briefly give an overview of a typical interaction with the Discovery Platform and motivate the need for the query infrastructure presented in this paper.

The Open PHACTS Discovery Platform exposes a domain specific web service API to a variety of end user applications. Two key groups of methods are provided: (i) *resolution methods* that resolve a user entry, e.g. text for the name of a compound, to a URI for the concept; and (ii) *data retrieval methods* that extract the data from the underlying datasets which have been cached into a single triplestore. A typical interaction first uses a resolution method to obtain a URI for the user input and then uses that URI to retrieve the integrated data.

Each data retrieval method corresponds to a SPARQL query that determines which properties are selected from each of the underlying data sources and the alignment between the source models. However, the query is parameterized with a single URI that is passed in the method call. Prior to execution over the triplestore this single URI needs to be expanded to the equivalent identifiers for each of the datasets involved in the query. The next section provides details of the query infrastructure that enables this functionality through stand-off mappings that can be varied depending upon the context of the user.

3 Identity Mapping and Query Expansion Services

To support the Open PHACTS Discovery Platform, we have developed a Query Expansion service that replaces data instance URIs in a query with "equivalent" URIs. The equivalent URIs are decided by the Identity Mapping Service (IMS). Note that equivalence is assumed to be with respect to some context. Obtaining and managing contextual links, and their effects, are not discussed in this paper. Instead we focus on ensuring the query infrastructure can return results in interactive time, even when there are many equivalent URIs.

3.1 Identity Mapping Service

The IMS extends the BridgeDB database identifier cross-reference service [14] for use in a linked data system. Given a URI, the IMS returns a list of equivalent URIs drawn from the loaded VoID linksets. A VoID linkset provides a set of links that relate two datasets together with associated provenance information [1].

Multiple namespaces can be used for a dataset, e.g. UniProt is available at `http://www.uniprot.org/uniprot/` and `http://purl.uniprot.org/uniprot/`.

```
1  PREFIX chemspider: <http://rdf.chemspider.com/#>
2  PREFIX sio: <http://semanticscience.org/resource/>
3  SELECT DISTINCT ?inchikey ?molformula ?molweight
4  WHERE {
5      GRAPH <http://rdf.chemspider.com> {
6          <http://rdf.chemspider.com/2157> chemspider:inchikey ?inchikey .
7      }
8      GRAPH <http://linkedchemistry.info/chembl> {
9          <http://rdf.chemspider.com/2157> sio:CHEMINF_000200 _:node1 .
10         _:node1 a sio:CHEMINF_000042; sio:SIO_000300 ?molformula .
11         OPTIONAL {
12             <http://rdf.chemspider.com/2157> sio:CHEMINF_000200 _:node2 .
13             _:node2 a sio:CHEMINF_000198; sio:SIO_000300 ?molweight .
14 }   }   }
```

Fig. 1: Simplified query for the compound Aspirin generated by the compound information API call, i.e. with a single URI for all statements.

Rather than duplicate the mappings across each of these equivalent namespaces, we support the mapping of namespaces at the dataset level.

The IMS exposes a web service API on top of a MySQL database. The database contains the mappings together with the metadata available about the linksets from which they were read. The source code is available from `http://github.com/openphacts/BridgeDB` and the service is available through `http://dev.openphacts.org/`.

3.2 Query Expansion Service

The Query Expansion service takes as input a SPARQL query and expands all of the data instance URIs appearing in that query with equivalent URIs drawn from the IMS. It is assumed that the query has been written with regard to the schema of each of the datasets involved but does not have appropriate mappings for the resource URIs. For example, consider a query such as the one given in Fig. 1; a simplified version of the Open PHACTS Discovery Platform compound information query which retrieves data from two datasets. The query has been instantiated with the URI from the ChemSpider database for *"Aspirin"*, *viz.* `http://rdf.chemspider.com/2157`, which was returned by the resolution step (Section 2). The query expander is responsible for replacing each instance URI with "equivalent" URIs retrieved from the IMS, in this case `http://linkedchemistry.info/chembl/molecule/m1280`.

The query expander has implemented two equivalent expansion strategies which can be selected between by providing a parameter. The first replaces each instance URI by a variable and introduces a `FILTER` statement to consider all equivalent URIs. For the example query, line 6 would be replaced with

```
?uri1 chemspider:inchikey ?inchikey  .
FILTER(?uri1 = <http://rdf.chemspider.com/2157> || ?uri1 =
    <http://linkedchemistry.info/chembl/molecule/m1280>)
```

The second approach introduces `UNION` clauses for each of the equivalent URIs. For the example query, line 6 would be replaced with

```
{ <http://rdf.chemspider.com/2157> chemspider:inchikey ?inchikey . }
UNION { <http://linkedchemistry.info/chembl/molecule/m1280>
        chemspider:inchikey ?inchikey . }
```

Note that the information contained in the results of queries constructed based on these expansion strategies is the same, even though strictly speaking the queries will produce different bindings sets.

Data integration queries consist of a collection of statements over multiple data sources. The *localise SPARQL subpatterns* heuristic, where statements are grouped into graph blocks according to their data source, results in more performant queries [15]. For instance URI expansion, if the heuristic is not followed then each statement is expanded independently generating an exponential number of alternatives which generally will not yield query answers. Additionally, if there is a one-to-many instance mapping, then the source URI will be expanded to two separate target URIs in the same dataset in two different statements. This is not the evaluation semantics that the user desires. Queries written according to the heuristic can avoid this shortcoming by the expansion service employing two optimisations. First, given a mapping between the graph block name and the underlying data source, the query expander can limit the set of expanded URIs to those that occur in that dataset. Second, when there are many URIs for a given graph block, these are bound once for the block rather than on a per statement basis, i.e. if there are y equivalent URIs you will get y bindings for the graph block rather than x^y, where x is the number of statements in the graph block. For instance, when expanding the statement on line 6 of Fig. 1 we only need to insert the ChemSpider URI, not the equivalent ChEMBL one as it does not appear in the ChemSpider dataset. Note that since there is only one ChemSpider URI we do not need to employ a filter or union block and directly insert the corresponding URI. As shown in [15], eliminating `FILTER` and `UNION` blocks results in more performant queries. These optimisations are essential for the complex queries used in the Open PHACTS Discovery Platform.

The code for the query expansion service is available from `https://github.com/openphacts/QueryExpander` and a deployment is accessible from `http://openphacts.cs.man.ac.uk:9090/QueryExpander/`.

4 Evaluation

This section describes our experimental evaluation. We compare the query evaluation time of using stand-off mappings through the query expansion service with two baseline approaches. The evaluation investigates whether the use of a query expansion service introduces undue performance costs.

4.1 Experimental Design

The two URI instance expansion strategies presented in this paper (filter-by-graph and union-by-graph) are compared with the performance of the corre-

sponding queries when all the URIs are directly inserted in the query (perfect-URIs) and the linked data form of the query where link statements are included in the query (linked-data). That is, for the example query in Fig. 1 the statement

```
<http://rdf.chemspider.com/2157> skos:exactMatch ?chemblid .
```

is added to the where clause and the instance URIs in the ChEMBL graph replaced with the variable `?chemblid`. We compare the performance of calling the query expansion service and executing the resulting query with directly running the baseline queries against the same triplestore. As a consequence we anticipate that the query expander queries will be slower since they have to call the query expansion service before being executed over the triplestore. The queries, datasets and scripts used in the evaluation are available from `http://openphacts.cs.man.ac.uk/qePaper/`.

Queries and Datasets. The queries used are adapted from the Open PHACTS Discovery Platform API[7] methods *"Compound Information"*, *"Compound Pharmacology Paginated"*, *"Target Information"*, and *"Target Pharmacology Paginated"*. The only differences between the queries used in this paper and the corresponding Discovery Platform API methods are that pagination is not considered, and `CONSTRUCT` blocks are replaced with `SELECT` statements. The queries are characterised by their use of graph blocks to group statements by source (i.e. following the *localise SPARQL subpatterns* heuristic); involving a large number of optional statements (between 2 and 15); and returning a large number of properties (between 12 and 21). For comparison purposes, we also used simplified forms of the *"compound information"* and *"target information"* queries that returned between 3 and 6 variables and at most one optional statement.

The data used corresponds to RDF dumps of ChEMBL version 13 [19], ChemSpider [16], ConceptWiki[8], and DrugBank [17]. The datasets mainly describe two types of resource: chemical compounds and targets (e.g. proteins). Additionally, a third type of resource is the interaction between a compound and a target which is used to answer the two pharmacology queries. In total, the data contains 168,783,592 triples, 290 predicates and are loaded in 4 separate named graphs (one per dataset).

Linksets have been separated out from the datasets and relate the concepts across the datasets. In total, there are five linksets providing 2,114,584 links. Note that the linksets were loaded into the evaluation triplestore to enable the linked-data evaluation, and represent one equivalence context.

Computational Environment. The experiments were conducted on a machine with 2 × Intel 6 Core Xeon E5645 2.4GHz, 96GB RAM 1333Mhz, and 4.3TB RAID 6 (7 × 1TB 7200rpm) hard drives. The Virtuoso Enterprise triplestore[9], version 07.00.3202 was used for the experiments.

[7] `https://dev.openphacts.org/` accessed July 2013
[8] `http://ops.conceptwiki.org/` accessed July 2013
[9] `http://virtuoso.openlinksw.com/` accessed July 2013

Evaluation Framework. The experimental evaluation was conducted using the Manchester University Multi-Benchmarking framework (MUM-Benchmark)[10] [2], which extends the Berlin SPARQL Benchmark [3] by allowing a query workload to be defined and run over an arbitrary dataset. The benchmark consists of two phases; a generation phase and an evaluation phase. Only the evaluation phase was timed. We ran 35 repetitions of the benchmark.

During the generation phase three versions of the queries were instantiated from the template queries with a randomly selected ConceptWiki URI of the correct type – compound or protein – for the query. This corresponds to the resolution step of the interaction with the Open PHACTS Discovery Platform (Section 2). During this phase, the perfect-URIs queries used the IMS to discover the equivalent URI for each graph block and these were inserted in the query. The other queries only contained the ConceptWiki URI.

During the evaluation phase the queries were executed either directly over the triplestore endpoint (perfect-URIs and linked-data) or through an endpoint that first called the query expansion service and then ran the resulting query (filter-by-graph and union-by-graph). Each run consisted of 10 warm-up evaluations of the query before timing the execution of 50 evaluations and reporting the average evaluation time.

4.2 Results

Due to space limitations, we are unable to present all of the results from the experiments. However, the raw experimental results and the generated graphs are available from [5].

In each of the graphs we use red to present the perfect-URIs baseline, blue for the linked-data baseline, green for the filter-by-graph expansion strategy, and orange for the union-by-graph strategy. The black dashes depict the number of query results returned. Note that the compound and protein information queries (queries 1, 2, 4 and 5) are expected to return a single result while the pharmacology queries (queries 3 and 6) have a variable number of results dependent upon the number of target/compound interactions.

Fig. 2 presents the average query execution time for each strategy over 35 random seed values focusing on the first five queries. The results show that the query expansion strategies are generally slower than our baselines as expected. However, in the worst case on query 2 (complete compound information) the query execution time for the filter-by-graph approach is 0.030 seconds, which is below human perception [6], and is compared to 0.006 seconds and 0.011 seconds for the perfect-URIs and linked-data baselines respectively. For query 6 (protein pharmacology, not shown due to clarity of presentation) the linked-data baseline performs significantly worse than all of the others; an average query execution time of 124 seconds compared to 6 seconds for each of the others.

In the compound information (query 2) and target information (query 5) queries, the linked data baseline produced two answers instead of the expected

[10] `https://code.google.com/p/mum-benchmark/` (revision 29) accessed July 2013.

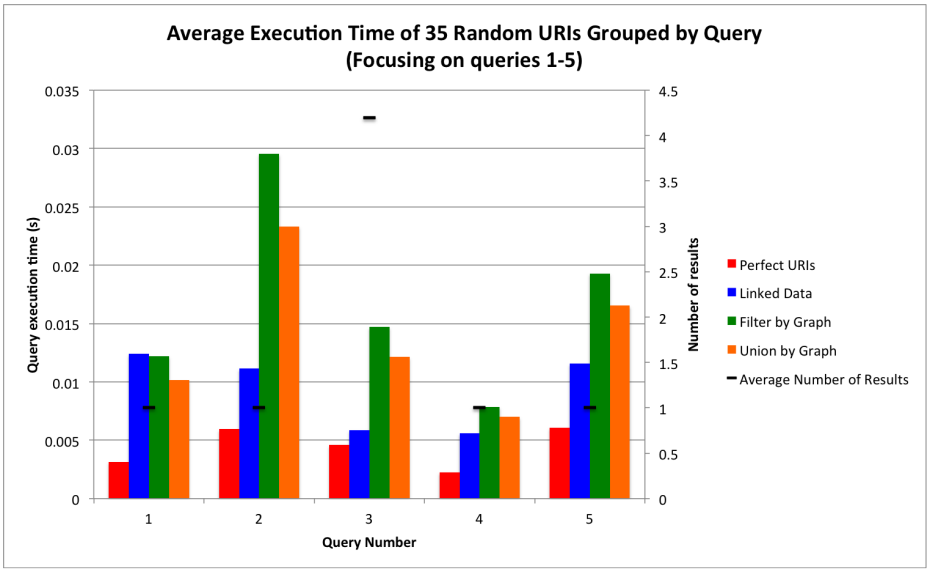

Fig. 2: Average query execution time of 35 random seed URIs grouped by query: only the results of queries 1-5 are shown. (Full results can be found at [5].)

single answer in ten and 22 cases respectively[11]. The cause for the two answers results from the DrugBank graph block being contained in an `OPTIONAL` clause. Due to the inclusion of the linking statements of the form

```
<http://rdf.chemspider.com/2157> skos:exactMatch ?chemblid .
```

in the linked-data query and the evaluation semantics of SPARQL this means that two results are obtained when the DrugBank URI is bound. This binding does not take place in the other forms of the query as they do not contain the link statement. However, this extra result does not significantly affect the evaluation time of the linked-data baseline. For the two simplified forms of these queries, queries 1 and 4, the linked-data baseline has more variation in its performance and was slower than the expansion strategies in runs 17 and 5 respectively.

Fig. 3 presents the query execution times for the compound pharmacology query, ordered by the number of results returned (shown by the black dashes). For all of the approaches, as the number of results increases the query execution time increases. In the majority of cases the linked-data baseline performed almost as well as the perfect-URIs baseline.

Fig. 4 presents the results for the protein pharmacology query (query 6), ordered by result size. The linked-data results as well as the results for runs 23 and 35 are ommitted for clarity of presentation. The linked-data approach performs exceptionally poorly for this query (taking at least 100 seconds). We suspect this is due to linking to the compound name for each interaction with the

[11] Not reflected in Fig. 2.

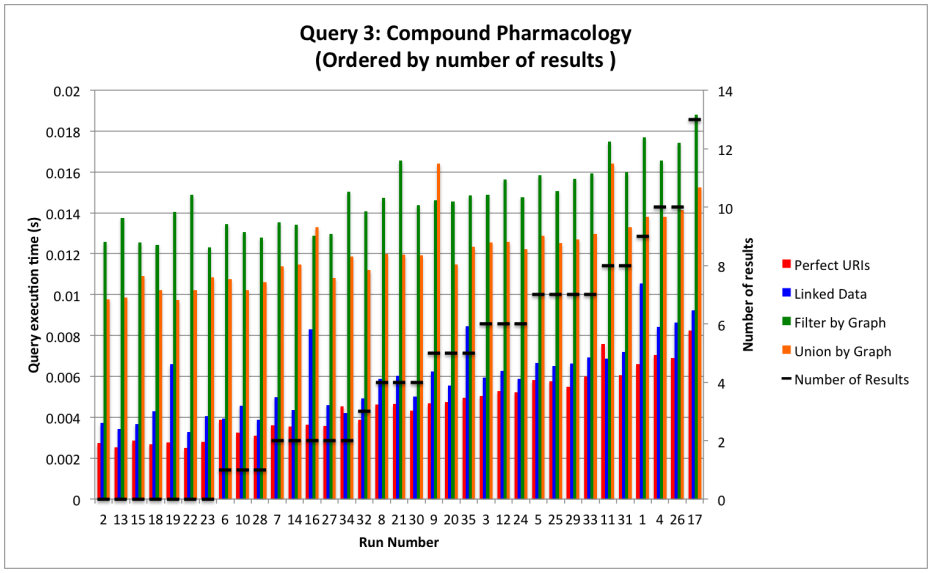

Fig. 3: Execution times for the compound pharmacology query ordered by result size.

seed protein. On runs 23 and 35, all four approaches took about 106 seconds to return 847 and 746 answers respectively. The expansion strategies have similar performance to the perfect-URIs baseline. Overall the results show a correlation between the result set size and the evaluation time.

4.3 Discussion

The performance measures show that in general the query expansion strategies are slower than both the perfect-URIs and linked-data baselines. This is due to the query expansion strategies first calling the query expansion service to expand the queries and then executing the resulting queries, whereas the baseline approaches directly executed the queries over the RDF store. The average difference to the perfect-URIs baseline for both query expansion approaches is 0.02 seconds. This is particularly evident in the results from the two pharmacology queries (Figs. 3 and 4) where large results sets are returned. In these queries, the time taken to return the answers dominates the execution time and thus the cost of URI expansion is mitigated. We note that the overall execution time of the query expansion queries is below the levels of human perception which are 0.05 to 0.2 seconds [6].

Of the two query expansion strategies, the union-by-graph strategy outperforms the filter-by-graph strategy in almost all cases. However, as shown in [15] the difference between processing `FILTER` and `UNION` clauses is dependent upon the triplestore used and it was shown that Virtuoso performs better with `UNION`. These experiments corroborate that result.

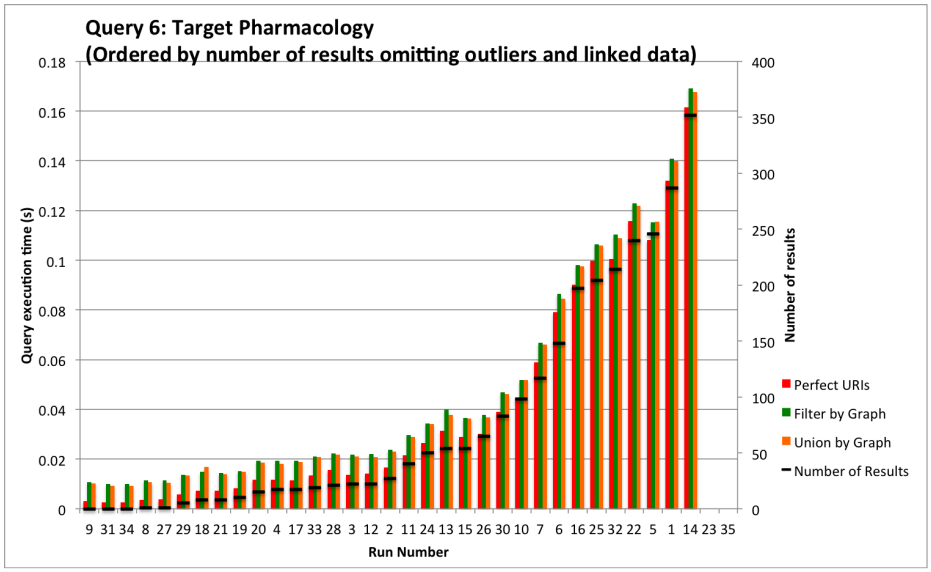

Fig. 4: Execution times for the protein pharmacology query: omitting the linked data approach and runs 23 and 35. (Full results can be found at [5].)

5 Related Work

The ultimate goal of our work is to integrate data from multiple data sources. Data integration has been widely studied both in the relational database community and in the semantic web community [8]. Integration systems expose a single view of the world to users and require the work of a domain expert to interrelate the datasets to be integrated. Dataspaces [11] aim to lower the up front cost by starting with rough relationships that can be refined automatically through user feedback. Our approach is similar in that the integration is achieved through queries and the relationships between datasets is captured in our global-as-view queries. However we enable different views of the data to be shown based on the equivalence relationships for the instance URIs in the query.

Linked data [13] is another approach to integrate data which uses instance level relationships. Links are embedded in the data that relate the entities in one dataset with those in another. The `owl:sameAs` predicate is widely used as the linking relationship, even though the intended interpretation does not match the OWL semantics. In fact, Halpin et al. showed that it is hard to correctly characterise the semantic relationship between two entities [12]. Moreover, the embedding of equivalence links in the data imposes a single view of the linked data web. In our work, we enable different views to be configured at query time, by requesting different sets of equivalent URIs from a mapping service. In this way we avoid the need to misuse `owl:sameAs`, or other equivalence predicates, and support the ability to adaptively turn linksets on and off at query time.

Correndo et al [7] proposed an approach for rewriting a SPARQL query expressed over one ontology to a query over a second ontology which also rewrote the instance URIs in the query. This is achieved through ontology alignments and the sameas.org service. They identified the problem that a rewriting does not hold in all contexts, which is the motivation for our query expansion infrastructure. The principal difference between their approach and ours is that they completely rewrite the query into a query over the second source whereas we leave the global-as-view query unchanged except for the inclusion of equivalent instance URIs. This is because we focus on instance level integration rather than schema integration. An important difference is that this work provides performance details on queries produced from real-world datasets.

6 Conclusions

In this paper, we presented a query infrastructure to enable the use of co-referent URIs drawn from stand-off mappings. This allows the equivalence of URIs to be decided by a mapping service such as BridgeDB or sameas.org, rather than embedded in the data. Such a service is required to support contextualised mappings that provide multiple views over linked data at query execution time, thus enabling the use of scientific lenses [4]. The performance of two strategies, implemented as a query expansion service on top of a BridgeDB identity mapping service, were compared with two baselines. While the query expansion and execution performance times were slower (in general) than the baselines, the difference in response times was still below the threshold of human perception. The extra time can be accounted for by the need to call the expansion service to expand the query with equivalent URIs. As future work we will be enabling the IMS to support different views over the data through the application of scientific lenses [4]. We will also investigate the generation of context dependent linksets from information available in existing datasets.

Acknowledgements The research has received support from the Innovative Medicines Initiative Joint Undertaking under grant agreement number 115191, resources of which are composed of financial contribution from the European Union's Seventh Framework Programme (FP7/2007- 2013) and EFPIA companies' in kind contribution, and the UK EPSRC myGrid platform grant (EP/G026238/1). We would like to thank Bijan Parsia for the discussions on SPARQL semantics and Samantha Bail for her help with the MUM-Benchmark.

References

1. Alexander, K., Cyganiak, R., Hausenblas, M., Zhao, J.: Describing linked datasets with the void vocabulary. Note, W3C (March 2011), `http://www.w3.org/TR/void/`
2. Bail, S., Alkiviadous, S., Parsia, B., Workman, D., van Harmelen, M., Gonçalves, R.S., Garilao, C.: Fishmark: A linked data application benchmark. In: SSWS+HPCSW. pp. 1–15. CEUR Workshop Proceedings (2012)

3. Bizer, C., Schultz, A.: The berlin sparql benchmark. International Journal on Semantic Web and Information Systems 5(2), 1–24 (2009)
4. Brenninkmeijer, C.Y.A., Evelo, C., Goble, C., Gray, A.J.G., Groth, P., Pettifer, S., Stevens, R., Williams, A., Willighagen, E.L.: Scientific lenses over linked data: An approach to support task specific views of the data. a vision. In: LISC2012. CEUR (2012)
5. Brenninkmeijer, C.Y.A., Goble, C., Gray, A.J.G., Groth, P., Loizou, A., Pettifer, S.: Complete results for including co-referent uris in a sparql query. http://dx.doi.org/10.6084/m9.figshare.701203 (2013)
6. Card, S.K., Moran, T.P., Newell, A.: The Psychology of Human-Computer Interaction. Lawrence Erlbaum Associates, Inc (1983)
7. Correndo, G., Salvadores, M., Millard, I., Glaser, H., Shadbolt, N.: Sparql query rewriting for implementing data integration over linked data. In: EDBT/ICDT Workshops (2010)
8. Doan, A., Halevy, A., Ives, Z.: Principles of data integration. Morgan Kaufmann (2012)
9. Euzenat, J., Shvaiko, P.: Ontology matching. Springer, first edn. (2007)
10. Gray, A.J.G., Groth, P., Loizou, A., Askjaer, S., Brenninkmeijer, C., Burger, K., Chichester, C., Evelo, C.T., Goble, C., Harland, L., Pettifer, S., Thompson, M., Waagmeester, A., Williams, A.J.: Applying linked data approaches to pharmacology: Architectural decisions and implementation. Semantic Web Journal To appear. http://www.semantic-web-journal.net/system/files/swj258_1.pdf
11. Halevy, A.Y., Franklin, M.J., Maier, D.: Principles of dataspace systems. In: PODS 2006. pp. 1–9. ACM (2006)
12. Halpin, H., Hayes, P.J., McCusker, J.P., McGuinness, D.L., Thompson, H.S.: When owl: sameas isn't the same: An analysis of identity in linked data. In: ISWC (1). pp. 305–320 (2010)
13. Heath, T., Bizer, C.: Linked Data: Evolving the Web into a Global Data Space. Morgan & Claypool (2011)
14. van Iersel, M.P., Pico, A.R., Kelder, T., Gao, J., Ho, I., Hanspers, K., Conklin, B.R., Evelo, C.T.A.: The bridgedb framework: standardized access to gene, protein and metabolite identifier mapping services. BMC Bioinformatics 11, 5 (2010)
15. Loizou, A., Groth, P.: On the formulation of performant sparql queries Submitted for publication. http://arxiv.org/abs/1304.0567
16. Pence, H.E., Williams, A.: ChemSpider: An Online Chemical Information Resource. Journal of Chemical Education 87(11), 1123–1124 (2010)
17. Samwald, M., Jentzsch, A., Bouton, C., Kallesoe, C., Willighagen, E., Hajagos, J., Marshall, M., Prud'hommeaux, E., Hassanzadeh, O., Pichler, E., Stephens, S.: Linked open drug data for pharmaceutical research and development. Journal of Cheminformatics 3(1), 19+ (2011)
18. Williams, A.J., Harland, L., Groth, P., Pettifer, S., Chichester, C., Willighagen, E.L., Evelo, C.T., Blomberg, N., Ecker, G., Goble, C., Mons, B.: Open PHACTS: Semantic interoperability for drug discovery. Drug Discovery Today (21-22), 1188–1198 (2012), 10.1016/j.drudis.2012.05.016
19. Willighagen, E.L., Waagmeester, A., Spjuth, O., Ansell, P., Williams, A.J., Tkachenko, V., Hastings, J., Chen, B., Wild, D.J.: The chembl database as linked open data. Journal of Cheminformatics 5(23) (2013), 10.1186/1758-2946-5-23