Balloon Fusion: SPARQL Rewriting Based on Unified Co-Reference Information

Kai Schlegel, Florian Stegmaier, Sebastian Bayerl, Michael Granitzer, Harald Kosch

University of Passau
Innstraße 43, 94032 Passau, Germany
forename.surname@uni-passau.de

***Abstract*—While Linked Open Data showed enormous increase in volume, yet there is no single point of access for querying the over 200 SPARQL repositories. In this paper we present Balloon Fusion, a SPARQL 1.1 rewriting and query federation service build on crawling and consolidating co-reference relationships in over 100 reachable Linked Data SPARQL Endpoints. The results of this process are 17.6M co-reference statements that have been clustered to 8.4M distinct semantic entities and are now accessible as download for further analysis. The proposed SPARQL rewriting performs a substitution of all URI occurrences with their synonyms combined with an automatic endpoint selection based on URI origin for a comprehensive query federation. While we show the technical feasibility, we also critically reflect the current status of the Linked Open Data cloud: although it is huge in size, access via SPARQL Endpoints is complicated in most cases due to missing quality of service.**

I. Introduction

Within the rise of the Semantic Web, the Web yielded from a more or less static and silo-based Web to an open Web of Data. Its vision is to provide an openly-accessible and machine-readable description of content, mediated by unified description semantics. Following this, Linked Open Data [1] is the de-facto standard to publish and interlink distributed data sets in the Web. The first wave of the Linked Open Data movement tackled the support in the creation of the envisioned Web of Data by producing a plethora of (open source) tools[1] to lift (legacy) data into the cloud. Besides all advantages and yet achieved results, however, the Linked Open Data cloud is mostly restricted to academic purposes due to unreliability of services and a lack of quality estimations of the accessible data. Statistics[2] of the Linked Open Data Cloud collected by the EU FP7 LOD2 project indicate that the cloud consist of approximately 700 data sets. The numbers further state that only half of the data sets are syntactically valid and only 50 are correctly exposed by a SPARQL endpoint. Regardless the pro-rata small amount of actual accessible Linked Open Data endpoints via a SPARQL interface, the Semantic Web community spots the need for an intelligent query federation service [2], [3]. While trying to establish a preferable efficient query federation, the question still remains, which subset of the Linked Open Data cloud retrieves meaningful information over a specific semantic entity. To overcome this task, Linked Open Data indices are being widely generated to locate the real information stakeholders in contrast to the manually specified interlinkings of single endpoints [4], [5].

The Balloon Fusion approach improves the aforementioned limitations by following a mediator-based architecture enabling a single point of access to a client application. The core contributions of this paper are as follows:

- *SPARQL-based crawling of co-reference information* of retrievable endpoints registered at *the Data Hub* platform[3].
- *Co-reference clustering* to enable an on-the-fly, intelligent and automatic query extension by using synonym URIs.
- *Algebra-based SPARQL 1.1 query rewriting* to ensure validity of the query at the syntactic level including an automatic federation functionality.

The outline of the paper is as follows: Section II introduces the overall scenario in which the proposed query federation technique is motivated. Together with related research in Section III this will serve as a foundation for the conceptional overview of the basic components given in Section IV. Section V as well as Section VI explain the main contributions, namely the co-reference index and the query transformation, in detail. The prototypic showcase of Section VII underlines the applicability of this approach and sheds light on the actual dataset used as index. Section VIII concludes the paper wit a summary of the proposed approach.

II. Information Discovery in the Web of Data

Following the main idea of the Linked Open Data project, data within the Linked Open Data community is highly interlinked, openly available, and can be queried separately on each endpoint. Harmonization at the modeling level is established due to unified description schemes, such as RDFS or OWL, which characterize a formal syntax and common data semantic. However, a convenient request considering the globally described data set is only possible with strong limitations. Given a specific entity, only Semantic Web experts may find all relevant information since those are distributed over several endpoints. A straightforward consumption of Linked Data is, among other things, hindered by the absence of a global view on all equivalent entities in the Linked Data cloud. Although utilizing URIs assure the identification of resources, a semantic entity can be referred by multiple overlapping,

[1] http://www.w3.org/wiki/ConverterToRdf
[2] http://stats.lod2.eu/
[3] http://datahub.io/

but different, URIs. When querying for a semantic entity, a user initially doesn't know which identifier for the semantic entity exists in a particular endpoint. As an example, one might know the URI of the DBpedia[4] resource of the Austrian ski jumper Thomas Morgenstern since it is the most common and domain independent endpoint in the Linked Open Data cloud: *http://dbpedia.org/resource/Thomas_Morgenstern*.

One could think of enlisting Freebase[5] for further information about Thomas Morgenstern with the same URI. Unfortunately, this leads to no results because the given URI does not exist in Freebase. Freebase has its own identifier for the same semantic entity, complicating an efficient querying or browsing inside the data graph. In linguistics, this issue is known as *co-reference issue* meaning multiple expressions refer to the same thing [6].

From the perspective of a Semantic Web user a central issue is how to automatically discover and query multiple endpoints that use different URI naming schemes. For that reason, we see the need for a discovery of relevant synonym URIs to enable an automatic integration in the query process in conjunction with an smart endpoint selection. An automatic query enhancement and intelligent routing to suitable endpoints fulfills the wish for an ease information access. That way, the fundamental idea of declarative programming of SPARQL would be reflected by specifying the characteristics of a solution to a problem, rather than specifying the steps that have to be followed to come to this solution. With a single point of access, the Linked Open Data cloud moves a step closer to the ultimate goal towards an openly accessible and unified data silo.

III. Related Work

Various literature explore the deployment of co-reference information in the context of the Semantic Web and therefore can be considered as related to our topic. An investigation of improper use of owl:sameAs [7] shows, that semantic entity identity is more complex than just applying owl:sameAs statements. Glaser,Jaffri and Millard outlined the co-reference problem in the Semantic Web including the need for a co-reference managing system and proposed a similar idea of indexing co-reference information, called co-reference resolution service (CRS)[6] [8], [9]. A quantitative analysis of owl:sameAs deployment and corresponding sameAs networks is discussed by Ding et al. [10], showing deep insight in the importance of co-reference and its usage in very large datasets. As a result a picture of co-reference deployment of the Linked Data community is presented. Utilizing a co-reference resolution service Correndo et al. perform a SPARQL query rewriting for ontology alignment between two different endpoints to achieve RDF dataset mediation at the level of noncompliant identifiers [11].

In the context of discovering and publishing of interlinking information, the Silk Framework[4] offers a declarative language for an easy generation of co-reference information between different data sources. SchemEX [12] represents an approach of stream-based indexing of schema knowledge (e.g type cluster, equivalence classes) used for locating of distributed data sources in the Linked Data Cloud.

The topic of SPARQL rewriting with automatic SPARQL federation is widely discussed. Hartig, Bizer and Freytag proposed an iterative query extension to execute SPARQL queries over the Web of Linked Data by following RDF links during execution time [2]. Exploiting formal mappings between different overlapping ontologies, Makris, Gioldasis, Bikakis and Christodoulakis invented an approach of SPARQL rewriting by the use of a mediator based system [13]. Resting on statistical information provided by VoiD descriptions [14], SPLENDID [5] builds a federation infrastructure with automatic endpoints selection with regard to predicate distributions in the data sources. Another transparent access to Linked Data is enabled with the FedX system [3], based on no prior knowledge about the data sources. FedX provides a full-fledged SPARQL endpoint with on-demand federation support. Multiple intermediate sub-queries about triple patterns define the involved endpoints and are aggregated to a final result. To overcome the issue of problematic data availability during query federation ANAPSID offers an adaptive query engine for SPARQL endpoints that adapts query execution scheduler to run-time conditions (e.g. blocking data source) [15]. Due to the number of different approaches, Rakhmawati and Hausenblas compared a selection of federated SPARQL frameworks in terms of performance [16].

IV. Conceptual Overview

The main objective of Balloon Fusion is offering a mediator service between a SPARQL client application and a set of actual Linked Open Data SPARQL endpoints to reveal a single point of access. To get a basic idea about the developed approach, a short overview over the central components is given. The main idea behind the query rewriting process is to *(i) extend the initial query by synonym URIs* and *(ii) address a suitable set of endpoints* to improve the result set, without changing the enclosed query semantics. The primary focus of this approach is to accomplish an immediate query rewriting without any on-demand analyze or check queries on Linked Open Data endpoints, which often ends in long-lasting latencies or timeouts. The rewriting should not be accompanied with a vast amount of time overhead. Ballon Fusion enables a direct forwarding of relevant subqueries to corresponding endpoints.

Figure 1 illustrates an overall end-to-end retrieval workflow. The first component of Balloon Fusion is the *Co-Reference Indexing* service. It crawls co-reference information of reachable SPARQL endpoints from the Linked Open Data cloud, performs equality clustering and stores the resulting cluster in a database for further analysis. The second component is the *Query Rewriting* service. Within this process, the query algebra is analyzed for all occurrences of entity identifiers. Those URIs are enriched with synonym URIs fetched from

[4] http://www.dbpedia.org/

[5] http://www.freebase.com/

[6] http://sameas.org/

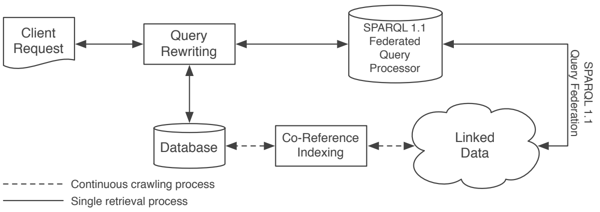

Fig. 1. Conceptual Overview

the database by reformulating the query and adding SPARQL 1.1 Federated Query[7] components based on the origin of the URIs. The rewritten query is then forwarded to a SPARQL 1.1 Federated Query Processor. In the prototype Jena ARQ[8] is utilized for this task and handles the communication with other Linked Data endpoints. In the following, the main components of the Balloon Fusion service are discussed in more detail.

V. Indexing Co-Reference Information

To overcome the issue of distributed portions of information in combination with a missing global view on Linked Data, Balloon Fusion aims at the aggregation of co-reference information of Linked Open Data endpoints to generate a simplified but global sub-graph. One of the main features of Ballon is that only basic "interlinking"-information are indexed whereas real "content"-information are fetched at query time from multiple endpoints.

Although a mature co-reference resolution service (CRS)[9] already exists, Balloon Fusion implemented a similar approach to fulfill all desired requirements on the retrieved data. Those are missing data provenance chains (which data source produced a relation) and the lack of cluster manipulation possibilities to enhance the overall quality and correctness.

The creation of the co-reference clusters leads to a bi-directional view on the co-reference relationships and is the result of a continuous indexing process of SPARQL endpoints. The complete crawling process is visualized in Figure 2. In contrast to the mentioned CRS, this approach considers only co-reference information from Linked Data SPARQL endpoints. The integration of RDF dumps or other data sources would result in an overhead because of mandatory manual activities leading to high maintenance efforts. Further, SPARQL indexing allows a periodically automatic update of the index and explicit data sources provide great traceability along with the needed provenance information.

The indexing process can be divided into several subparts. Fundamentally, the *Endpoint Manager* is aware of tethered SPARQL endpoints and starts a new indexing processes for a specific SPARQL endpoint by instantiating a corresponding task. Each *Index Process* is responsible for the querying of co-reference information from a SPARQL endpoint. The query asks for all subjects and objects which are associated by a special predicate. There exist several ways to indicate semantics equivalencies instead of just utilizing `owl:sameAs`. For the crawling process, we consider the following set of predicates:

- http://www.w3.org/2002/07/owl#owl_sameAs
- http://www.w3.org/2004/02/skos/core#exactMatch
- http://www.rkbexplorer.com/ontologies/coref#coreferenceData
- http://www.geneontology.org/formats/oboInOwl#hasExactSynonym

Without going into details on the technical point of view, the querying of those relations is a hard task due to retrieval limitations (e.g. maximum result size or quota) and responsiveness of the endpoints. Amongst others, the index process makes use of pagination to create the final result set iteratively, tries to exploit the maximum of the limitations to minimize the retrieval duration and pause the indexing when temporary bandwidth restrictions occur.

Within the complete process, a NoSQL database is used as persistence layer in order to provide scalability and high availability for huge data sets. The prototype is build on mongoDB[10]. Compared to other NoSQL stores, mongoDB is perfect for high read workload and occasionally write operations, which is inline with the workflow of the described process.

The next step includes the clustering of the retrieved co-reference information with respect to equality and dissolving transitivity or asymmetry of existing equal statements in the cluster. This means the aggregation of all related URIs to produce a single cluster containing multiple URIs, which represent the same semantic entity. For efficient processing, this step is highly parallelized and controlled by the *Cluster Manager*. The resulting clusters do not only contain a set of "equal" URIs (the *equivalence set*), but also information about the included raw input statements and source endpoints. These provenance information allow analytics, live elimination of user-depending untrusted endpoints and endpoint selection during the rewriting process. The equivalence set represents a bi-directional complete sub-graph containing the equivalence relationship between all enclosed URIs and forms the foundation for the following SPARQL transformation.

Early experiments indicated, that Linked Open Data sometimes match "unexpected" entities as equal ("false friends"). As an example the German city "Berlin" has a equal relationship with the U.S. state "New Jersey", which makes in that place no sense and results in a huge inhomogeneous merged cluster. The issue of incorrect identity or misused equality has been discussed in detail by Halphin et al. [7]. To improve the overall data quality Balloon Fusion currently performs an asynchronous post-processing for the equality cluster. If a cluster has a certain size, it will be checked for bi-connected graphs and possibly divides them into separated clusters, if there is only one connecting edge. This is based on the assumption that redundancy and multiple paths in the equality-graph indicates trust. Further it is envisioned to offer the possibility for a crowd-sourced cluster manipulation and

[7] http://www.w3.org/TR/sparql11-federated-query/

[8] http://jena.apache.org/documentation/query/

[9] http://sameas.org/

[10] http://www.mongodb.org/

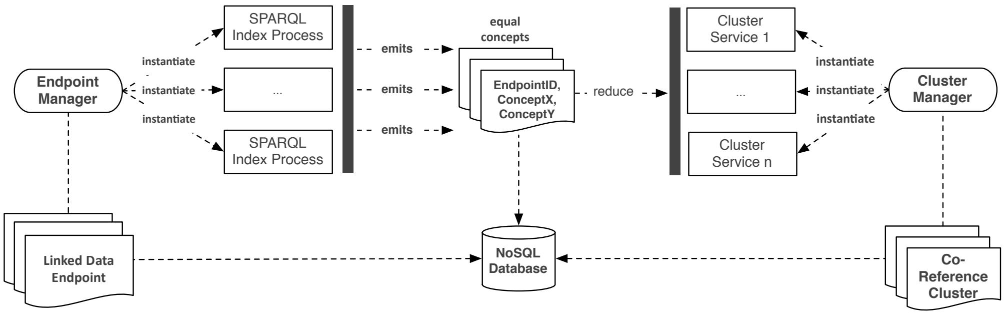

Fig. 2. Balloon Indexing Process

to perform association rule mining combined with a semantic similarity matching to avoid incorrect relationships.

VI. SPARQL Transformation

This section is focused on the SPARQL rewriting and endpoint selection. To illustrate the workflow, an example query transformation will be sketched. A SPARQL Query will be internally represented using the corresponding SPARQL algebra model [17], which defines the semantics of a SPARQL query execution. Basically, this algebra notes a syntax tree composed of different elements defining the well-known SPARQL query syntax elements: Basic Graph Pattern (BGP), Join, LeftJoin, Filter or Union. The proposed approach focuses on matching and replacing BGP elements containing a set of triple patterns. Hence, it consists only of leafs of the query tree and assures that the remaining query semantic stays untouched.

The first step of the SPARQL rewriting is a transformation of the initial SPARQL query to the corresponding SPARQL algebra. Listing 1 shows an example SPARQL query. In this query, all known predicates and objects for a given subject are queried. The transformed SPARQL algebra notation of the query is shown in Listing 2.

Listing 1. Example SPARQL query

```
SELECT ?p ?o WHERE {
 <http://vocab.semantic-web.at/AustrianSkiTeam/121> ?p ?o.}
```

Listing 2. SPARQL Algebra notation for example query

```
(project (?p ?o)
 (bgp
  (triple
   <http://vocab.semantic-web.at/AustrianSkiTeam/121> ?p ?o
)))
```

A. Determine synonym URIs

After the transformation step, the service analyses the SPARQL algebra in order to discover the occurrence of BGP-elements containing triple patterns. Those triple patterns are likewise analyzed to detect elements including a URI as subject or object of the statement. Only triples containing a URI are considered for transformation.

Given the above mentioned example query in SPARQL algebra notation (cp., Listing 2) the BGP containing the URI "http://vocab.semantic-web.at/AustrianSkiTeam/121" is the only candidate for transformation. Querying the co-reference index for synonym URIs results in a set of three URIs:

- "http://dbpedia.org/resource/Thomas_Morgenstern"
- "http://vocab.semantic-web.at/AustrianSkiTeam/121"
- "http://rdf.freebase.com/ns/m/08zld9"

As a practical improvement, we make use of a Linked Open Data principle in case that the Ballon Indexing Service did not yield any synonym results. The Linked Open Data rules recommend "when someone looks up a URI, provide useful information, using the standards (RDF or SPARQL)"[11]. HTTP offers a powerful mechanism to request a resource with a special result format known as *content negotiation*. Following this, one can directly look up the unknown URI by requesting RDF data and search the result for co-reference information.

B. Endpoint Selection

Based on the retrieved co-reference cluster appropriate SPARQL endpoints for an enhanced query federation setup can be exposed. Balloon Fusion identifies SPARQL endpoints using two different approaches. On the one hand Balloon Fusion makes use of the provenance information obtained by the indexing step to select endpoints, which are involved in the equivalence cluster composition (*provenance based selection*). If an endpoint contributed co-reference information for an entity, it might also have further information about the entity itself. On the other hand Balloon Fusion analyzes the retrieved synonym URIs to get additional endpoints by prefix and namespace matching, based on the assumption that the "owner" of a URI has further information about the resource (*namespace based selection*). Summarized, the origin of co-reference information and origin of synonym URIs are involved in the query execution to improve the result set.

In the following, let us assume the provenance information contains only one SPARQL endpoint (http://vocab.semantic-web.at/sparql/OpenData) as source for the co-reference relations for the given cluster. This endpoint stated that these three URIs are semantically equivalent by providing

[11] http://www.w3.org/DesignIssues/LinkedData.html

corresponding coreference relationships. In this light, Balloon Fusion could expose two different SPARQL endpoints for this BGP element:

- "http://vocab.semantic-web.at/sparql/OpenData"
- "http://dbpedia.org/sparql"

The first SPARQL endpoint is eligible because of the explained provenance information. Additionally the SPARQL endpoint "http://dbpedia.org/sparql" is qualified because of a positive namespace matching of synonym URI "http://dbpedia.org/resource/Thomas_Morgenstern". Namespace information are provided by *the Data Hub* for most of Linked Data endpoints.

C. Federated SPARQL Query Setup

Given the selected endpoints, we can establishing a federated querying, by using the W3C SPARQL 1.1 Federated Query extension[12]. With the supplemented SERVICE keyword, and a qualified federated query processor, a query or a partial query can be executed against a remote SPARQL endpoint. The query processor can be on a dedicated machine apart from the Linked Open Data and is only responsible for distribution and aggregation. The actual distributed queries do not contain SPARQL 1.1 features and hence don't require a special endpoint.

To integrate the exposed endpoints in the query, each distributable subquery has to be adapted for the corresponding endpoint, by determining suitable URIs for the target ontology.

Namespace based selected endpoints are requested with the matched synonym URI. Therefore adapting the BGP for namespace based selected endpoints can be achieved straightforward by replacing the original URI with the matched synonym URI.

In contrast, *provenance based selected endpoints* have to be requested with a union of the synonym URIs provided by this endpoint. In SPARQL semantics we can achieve this with a combination of SPARQL UNION blocks. The construct of UNION-combined synonyms will be subsequently referred to as *equivalence block*. In fact, there would be several ways in SPARQL to model this behavior, such as FILTER with IN operator or using the SPARQL 1.1 VALUES (former BINDINGS) feature[13]. But since the resulting queries should be accepted by many real Linked Open Data endpoints, we decided to only use basic, well implemented and widely adopted SPARQL features.

Listing 3. Example equivalence block for provenance based selected endpoint
```
(union
 (union
  (bgp
   (triple
    <http://vocab.semantic-web.at/AustrianSkiTeam/121> ?p ?o))
  (bgp
   (triple
    <http://rdf.freebase.com/ns/m/08zld9> ?p ?o)))
 (bgp
  (triple
   <http://dbpedia.org/resource/Thomas_Morgenstern> ?p ?o))))
```

The equivalence block for provenance based selected endpoint "http://vocab.semantic-web.at/sparql/OpenData" in the example would be structured like shown in Listing 3.

Each adapted BGP-element is encapsulated in a SERVICE element including the SPARQL endpoint itself. Now, the original statement in the initial query can be replaced by the union of the generated SERVICE components. The final SPARQL result of the example transformation can be found in Listing 4.

Listing 4. Result of example query transformation
```
SELECT ?p ?o WHERE {
 SERVICE <http://dbpedia.org/sparql> {
  <http://dbpedia.org/resource/Thomas_Morgenstern> ?p ?o.
 } UNION {
  SERVICE <http://vocab.semantic-web.at/sparql/OpenData> {
   {<http://dbpedia.org/resource/Thomas_Morgenstern> ?p ?o}
   UNION
   {<http://vocab.semantic-web.at/AustrianSkiTeam/121> ?p ?o}
   UNION
   {<http://rdf.freebase.com/ns/m/08zld9> ?p ?o}
}}}
```

It may also be the case that there are two URIs in the original BGP-element. In this case the cross product of the adapted URIs for each endpoint have to be constructed using UNION operators. Obviously, the hole transformation features a growth in the overall query size. Due to this fact, techniques for efficient pruning, the pre-selection of usable SPARQL endpoints as well as algebraic optimization techniques are an important task and will be focused in future work. Besides this, the condition and supported features of the underlying SPARQL endpoints are an interesting topic [18]. Currently we perform a check routine in terms of availability and latency for SPARQL endpoints. For this purpose, public available Mondeca SPARQL endpoints status [14] is integrated to identify and avoid unavailable or slow endpoints. Additionally, adapting the aforementioned SPARQL 1.1 VALUES feature instead of many UNION blocks would reduce the overall query structure and likely optimize the whole query processing. However, a proper and pervasive implementation of this feature must exist in order to take advantage of this improvement in practical use.

VII. Prototypic showcase & Details on dataset

Apparently, the Balloon Fusion service depends on a huge amount of pre-processed co-reference information to achieve a satisfying and immediate rewriting without on-demand queries on Linked Open Data endpoints. For this reason, we started to index the Linked Open Data based on the metadata which can be queried from *the Data Hub*. At the time of writing, we identified 237 Linked Open Data endpoints exposing a SPARQL endpoint and therefore serve as candidates for our indexing service. Unfortunately, only a small subset of endpoints were exploitable without manifesting issues. The initial expectation of huge amount of interlinked data that is available via many active endpoints is tarnished, because the real Linked Open Data shows up a poor quality of service. Probably this is based on a lack of long-term data curation as well as maintenance. Among direct solvable issues like HTTP redirects, there were

[12] http://www.w3.org/TR/sparql11-federated-query/

[13] http://www.w3.org/TR/sparql11-query/#inline-data

[14] http://labs.mondeca.com/sparqlEndpointsStatus/

many problems like incorrect URIs, unreachable SPARQL endpoints, access right issues or simply insufficient support of SPARQL, which led to exclusion of many endpoints. In sum, 125 endpoints had to be excluded and 112 endpoints were successfully queried for co-reference information. As a result, our dataset contains over 17.6M statements including 22.4M distinct URIs which represent co-reference information. Cluster computations showed, that this data results in about 8.4M "synonym"-groups. In total average each groups contain 2.68 different entities. To be traceable, all individual indexed data sets are available as open download in a zipped way[15].

A simplified prototype of our approach is deployed under http://purl.org/balloon/demo. Besides a demonstration of the presented workflow, it offers a SPARQL 1.1 capable endpoint, which achieves an automatic approach of query federation without initial knowledge of suitable Linked Open Data endpoints. Furthermore, all described services, e.g. querying for co-reference cluster for a given URI or resolving of SPARQL endpoints, are available as public HTTP-API.

VIII. Conclusion

This paper introduced Balloon Fusion, a service for creating a single point of access to the Linked Open Data cloud. Its main contributions are the SPARQL-based crawling of co-reference information, the co-reference cluster index itself as well as on-the-fly query rewriting and federation components. We spotlight, that Balloon Fusion pursues a direct and easy consumption of distributed data sources and is therefore peculiarly adapted for use cases like information browsing.

While creating the co-reference index, we encountered several issues in the current state of Linked Open Data cloud. Missing maintenance of endpoints over years as well as a lack of quality of service hinder the Linked Open Data cloud to fully unfold its entire potential. To reveal the *real* state of the cloud, the collected co-reference information will be deeply analyzed in future work. In this domain, core questions such as information distribution will be discussed as well as the mostly manually generated information on endpoints in terms of interconnections validated.

The demonstrated services, in detail the query rewriting component as well as the co-reference index, are not tied to a specific federation system. They follow a modular architecture and could be reused in arbitrary frameworks to establish an smart query federation. We envision to integrate this approach as add-on in other systems like the open source community project Apache Marmotta[16].

Acknowledgment

The presented work was developed within the CODE project (http://code-research.eu/) funded by the EU Seventh Framework Programme, grant agreement number 296150.

References

[1] T. Heath and C. Bizer, *Linked Data: Evolving the Web into a Global Data Space*, ser. Synthesis Lectures on the Semantic Web. Morgan & Claypool Publishers, 2011.

[2] O. Hartig, C. Bizer, and J.-C. Freytag, "Executing SPARQL queries over the web of linked data," in *Proceedings of the 8th International Semantic Web Conference*, 2009, pp. 293–309.

[3] A. Schwarte, P. Haase, K. Hose, R. Schenkel, and M. Schmidt, "Fedx: Optimization techniques for federated query processing on linked data," in *The 10th International Semantic Web Conference*, 2011, pp. 601–616.

[4] J. Volz, C. Bizer, M. Gaedke, and G. Kobilarov, "Silk–a link discovery framework for the web of data," in *Proceedings of the 2nd Linked Data on the Web Workshop*, 2009, pp. 559–572.

[5] O. Görlitz and S. Staab, "Splendid: Sparql endpoint federation exploiting VOID descriptions," in *Proceedings of the 2nd International Workshop on Consuming Linked Data, colocated with the 10th International Semantic Web Conference*, 2011.

[6] J. R. Hobbs, "Coherence and coreference," *Cognitive Science*, vol. 3, no. 1, pp. 67 – 90, 1979.

[7] H. Halpin, P. Hayes, J. McCusker, D. McGuinness, and H. Thompson, "When owl:sameas isn't the same: An analysis of identity in linked data," in *The 9th International Semantic Web Conference*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2010, vol. 6496, pp. 305–320.

[8] H. Glaser, A. Jaffri, and I. Millard, "Managing co-reference on the semantic web," in *WWW2009 Workshop: Linked Data on the Web (LDOW2009)*, 2009.

[9] A. Jaffri, H. Glaser, and I. Millard, "Managing uri synonymity to enable consistent reference on the semantic web," in *Identity and Reference on the Semantic Web 2008*, 2008.

[10] L. Ding, J. Shinavier, Z. Shangguan, and D. L. McGuinness, "Sameas networks and beyond: analyzing deployment status and implications of owl:sameas in linked data," in *Proceedings of the 9th International Semantic Web Conference on The Semantic Web*, ser. ISWC'10. Springer-Verlag, 2010, pp. 145–160.

[11] G. Correndo, M. Salvadores, I. Millard, H. Glaser, and N. Shadbolt, "Sparql query rewriting for implementing data integration over linked data," in *Proceedings of the 2010 EDBT/ICDT Workshops*, ser. EDBT '10. New York, NY, USA: ACM, 2010, pp. 4:1–4:11. [Online]. Available: http://doi.acm.org/10.1145/1754239.1754244

[12] M. Konrath, T. Gottron, S. Staab, and A. Scherp, "Schemex efficient construction of a data catalogue by stream-based indexing of linked data," *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 16, no. 5, 2012.

[13] K. Makris, N. Gioldasis, N. Bikakis, and S. Christodoulakis, "Ontology mapping and sparql rewriting for querying federated rdf data sources," in *On the Move to Meaningful Internet Systems*. Springer, 2010, pp. 1108–1117.

[14] K. Alexander, R. Cyganiak, M. Hausenblas, and J. Zhao, "Describing linked datasets - on the design and usage of void, the 'vocabulary of interlinked datasets'," in *Linked Data on the Web Workshop, in conjunction with 18th International World Wide Web Conference*, 2009.

[15] M. Acosta, M.-E. Vidal, T. Lampo, J. Castillo, and E. Ruckhaus, "Anapsid: an adaptive query processing engine for sparql endpoints," in *The Semantic Web*. Springer, 2011, pp. 18–34.

[16] N. A. Rakhmawati and M. Hausenblas, "On the impact of data distribution in federated sparql queries," in *Semantic Computing (ICSC), 2012 IEEE Sixth International Conference on*. IEEE, 2012, pp. 255–260.

[17] R. Cyganiak, "A relational algebra for SPARQL," *Digital Media Systems Laboratory HP Laboratories Bristol. HPL-2005-170*, 2005. [Online]. Available: http://www.hpl.hp.com/techreports/2005/HPL-2005-170.html

[18] C. Buil-Aranda, A. Hogan, J. Umbrich, and P.-Y. Vandenbussche, "Sparql web-querying infrastructure: Ready for action?" in *The Semantic Web ISWC 2013*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2013, vol. 8219, pp. 277–293. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-41338-4_18

[15] Available via FTP under ftp://moldau.dimis.fim.uni-passau.de/data/

[16] http://marmotta.incubator.apache.org/