

Linking the (un)linked data through backlinks

Michalis Stefanidakis
Ionian University, Dept. of Computer Science
Plateia Eleytherias
Corfu, 49100, Greece
mistral@ionio.gr

Ioannis Papadakis
Ionian University, Dept. of Archives
and Library Science
Ioannou Theotoki 72
Corfu, 49100, Greece
papadakis@ionio.gr

ABSTRACT

This paper proposes a framework capable of enhancing the interlinking between Linked Open Data – LOD repositories. Such interlinking is achieved through the realization of backlinking. Backlinking refers to the consistent management of the references that are made to a resource within a LOD repository from other repositories through their triples.

Categories and Subject Descriptors

H.3 [Information Systems Applications]: Information search and retrieval, Information Storage, On-line information Services

General Terms

Algorithms, Design

Keywords

Linked data, LOD, backlinks, interlinking

1. INTRODUCTION

The most popular way of information dissemination and exchange around the world is beyond any doubt the web. Information on the web appears as computer files that are grouped together in web pages. Web pages are linked together via hyperlinks. Users are able to move from one page to another by following the outgoing hyperlinks that are placed by web authors. Unfortunately, web authors are unaware of the hyperlinks that reference their pages (i.e. backlinks), not giving their visitors the opportunity to navigate to other possibly interesting pages.

The lack of backlinking functionality has been identified from online community sites (e.g. blogs) that have accordingly developed a notification mechanism in order to notify their users of possible backlinks by using pingbacks or trackbacks. In typical pingback-enabled blogs, a backlink is rendered in the feedback area of a target post together with

the title and a short text excerpt of the source resource. Such systems are capable of efficiently managing backlinks for their users. However, each community site has its own methodology in handling backlinks and therefore interoperability among different community sites does not exist.

In this paper, it is argued that LOD should learn from the web pages reality and accordingly provide bidirectional interlinking between remote resources. Along these lines, a framework is proposed capable of storing the references that are attributed to a local resource from remote LOD triples. It is argued that the proposed framework could be employed from various LOD-related services wishing to exploit bidirectional interlinking between LOD repositories.

2. CURRENT INTERLINKING STATUS

A wishful thinking about the concept of linked data is the ability to start searching from a LOD repository and seamlessly expand the search to other LOD repositories that possibly contain relevant information.

The first step towards the realization of this functionality is the employment of many namespaces within a single LOD repository. In this direction, apart from manual assignment, some efforts try to automatically match local resources with their identical ones located in remote LOD repositories (e.g. [2]).

Some research efforts that are targeted towards aggregative LOD services employ the notion of a catalog of remote endpoints, in order to broadcast a SPARQL query about a given resource (e.g. [3]). Others rely on LOD crawlers in order to localize parts of the entire Web of Data – WoD for further processing [1].

In an effort to mimic the online community sites' pingback mechanisms, the authors of [5] suggest that whenever a reference to a remote resource is made from a local LOD repository, the corresponding remote server should be notified and update its underlying repository with accordingly produced triples. Such an approach depends on the fact that a LOD repository introducing a resource will grant write access to remote repositories that reference such resource. Moreover, LOD repositories with 'popular' resources could face scalability issues, since the volume of their data depends on the references that are made to their resources.

The authors of [4] rely on a crawling service¹ in order to discover backlinks within the linked data of the UK Public Sector Information domain. However, depending on a crawling service to index all the resources that are available within

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

WIMS '11, May 25-27, 2011 Sogndal, Norway

Copyright 2011 ACM 978-1-4503-0148-0/11/05 ...\$10.00.

¹www.sameas.org

the WoD, merely reproduces the traditional erroneous Web paradigm where large-scale search engines decide which information should be indexed and accordingly accessed from their clients.

In the next section, a framework is presented capable of realizing the notion of bidirectional interlinking between resources in remote LOD repositories in a distributed manner.

3. PROPOSED APPROACH

The proposed solution tries to enhance the interlinking of resources stored in remote LOD repositories. The approach described herein follows a set of basic principles:

- Each LOD repository stores triples that contain its own resources (the “defined” resources) and resources that are defined in remote repositories (the “referenced” resources). A query about a resource will always be answered by the repository that defines the resource. The reply may contain augmented information based on triples stored in repositories where the resource is also referenced.
- In case of a repository referencing a remote resource (“forward link”), additional data should indicate its provenance: this entails not only the URI prefix but also the access point (e.g. SPARQL endpoint) of the remote repository that defines the corresponding resource.
- Symmetrically to the above, the LOD repositories should be able to consistently store the incoming references of the resources they define (“backlink”).
- Finally, the above described forward links and backlinks should ideally be created and maintained in a fully automated manner. This requires a framework accompanying each repository for the reception and storage of backlinking notifications.

3.1 Framework

For each distinct resource within a LOD repository, the following information is stored in an adequate indexing structure, called thereafter *the registry*:

1. For each defined resource in the registry, a table of backlinks is created, which stores the SPARQL endpoints of all repositories that reference the resource. This way, each repository knows the number of references of its defined resources and the way to access such references.
2. For each referenced resource in a LOD repository, the SPARQL endpoint of the remote repository where this resource is defined, is stored as a forward link.

The decision to attach a SPARQL endpoint to each resource is crucial to the operation of the proposed framework: queries may return rich responses if data from individual repositories are combined, crawled or aggregated. All these operations are served via the repositories’ SPARQL endpoints.

In figure 1, the typical interlinking between LOD repositories is illustrated as forward links whereas the proposed enhanced interlinking is illustrated as backlinks.

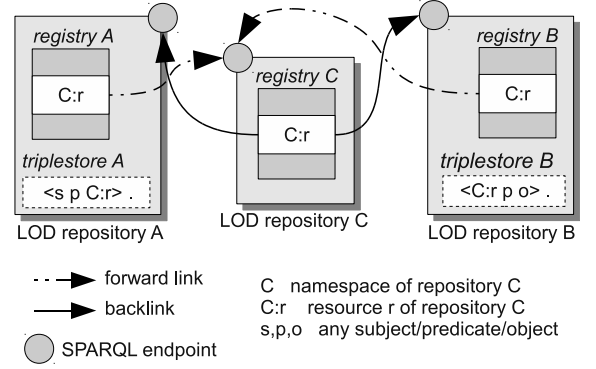


Figure 1: Interlinking scheme.

3.2 Backlinking process

The event of a backlink notification is triggered when a triple containing a referenced resource is created in a LOD repository. Upon creation of such a triple, two things happen: a) the referenced resource is inserted into the local registry together with the SPARQL endpoint of the remote LOD repository that defines the referenced resource, and b) the remote LOD repository is notified that a reference is made to one of its resources. Such notification is realized as an HTTP GET message (figure 2) containing the URI of the resource and the SPARQL endpoint of the LOD repository that referenced the resource.

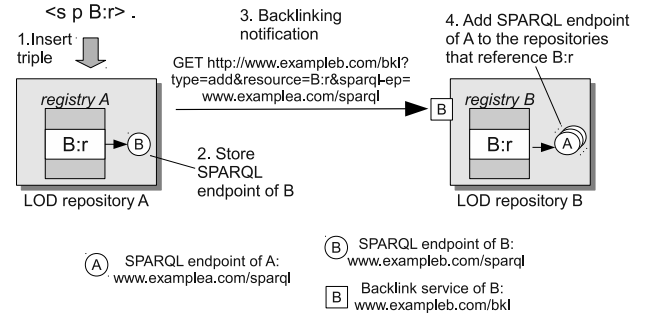


Figure 2: The backlinking process.

All LOD repositories maintain a service capable of receiving messages containing references to their defined resources. Upon reception, and given that the LOD repository’s policy accepts the message, the resource is matched against the entries of its local registry. In case of a successful match, the LOD repository stores the SPARQL endpoint of the remote repository that referenced the resource.

3.3 Usage scenarios

Backlinks can drastically improve interlinking between LOD repositories. Based on the proposed framework, a number of possible usage scenarios could be defined.

Along these lines, it is argued that the SPARQL DESCRIBE aggregative query would benefit from the proposed approach. When issuing a DESCRIBE query about a resource, a response containing rich information is anticipated. Currently, the LOD-cloud² gives the opportunity to LOD

²www.lod-cloud.net

applications to harvest data from various repositories. This can be achieved via crawling or dynamic aggregation of triples in real time. Such a process is inherently slow since queries must propagate from repository to repository, as interlinking facts appear. Moreover, the visiting sequence of the repositories greatly affects the process, as a query may reach a dead-end without being able to move on, even if further interlinking facts exist in others, yet unreachable, repositories.

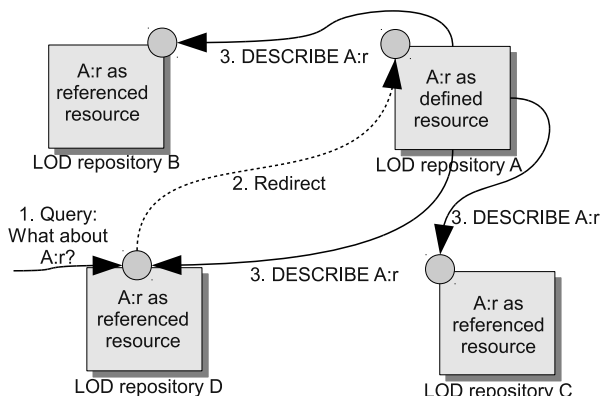


Figure 3: The SPARQL DESCRIBE aggregative query.

The proposed framework provides efficient paths of query execution: triples about a repository’s defined resources are always one hop away, via the stored set of remote SPARQL endpoints referencing the resource. In the case of a query about a referenced resource, the query is addressed to the SPARQL endpoint of the repository that defines the resource (figure 3). The bidirectional linking of repositories eliminates the ‘dead-ends’ in queries and consequently augments the richness of the corresponding results. Finally, the fact that each repository knows the number of references that are made to the defined resources, provides the opportunity to rank the retrieved resources according to their popularity.

4. EXAMPLES

In order to demonstrate the mechanism and the eventual usefulness of the proposed framework, a part of the LOD-cloud is studied with respect to the interlinking between repository nodes. This set of repositories is depicted in figure 4. Each node constitutes a repository in a loose definition: some nodes provide SPARQL endpoints, some other data dumps only, while a third category provides the bare minimum service of dereferenceable URIs to RDF content. The nodes under study (LCSH subject headings³, lobid bibliographical resources⁴ and DnB, the Linked Data service of the German National Library⁵) provide full RDF data dumps, facilitating thus the complete analysis of their underlying URIs. Shaded nodes in figure 4 are referenced by the datasets studied and, while sampled for RDF content, were not fully analyzed. In particular, shaded nodes with dotted borders (STITCH Rameau subject headings⁶,

lobid-organizations⁷, Dewey Decimal Classification Linked Data⁸ and Virtual International Authority File – VIAF⁹) do not provide any means of publicly acquiring their full RDF datasets as dumps. On the other hand, dbpedia¹⁰ does provide full RDF dumps but only its SPARQL endpoint is used in this study.

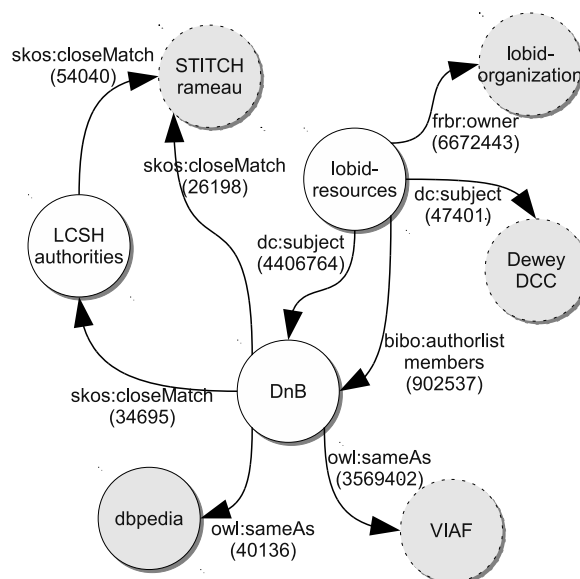


Figure 4: The LOD-cloud subset of study.

What is clearly visible in figure 4 is that links between LOD repositories present a significant semantic richness, far beyond the expressiveness of the widely employed *owl:sameAs* relation. The repositories under study contain bibliographical data, thus they express their external relations in relevant vocabularies (Dublin Core – dc¹¹, Bibliographic Ontology – bibo¹² and the Functional Requirements for Bibliographic Records – frbr¹³). Random sampling of the RDF content of the repositories adjoining the studied ones, reveals extended usage of identity constructs, such as the *skos:closeMatch* and the newer *umbel:isLike*¹⁴ relations. As the LOD-cloud grows, the expressiveness of the interlinking will become constantly richer. For this reason, scalable back-linking mechanisms should be relation-agnostic; there is no means to predict future interlinking relations. Mechanisms based solely on the *owl:sameAs* or *rdf:seeAlso* constructs will need to adapt in order to capture the evolving LOD interlinking.

At this point it should be noted that the framework proposed in this paper dutifully records the backlinks but does not interpret the semantics of the latter, as this is a task of the yet-to-come applications that will be using the framework.

⁷<http://lobid.org/de/organisation.html>

⁸<http://dewey.info/>

⁹<http://viaf.org/>

¹⁰<http://dbpedia.org/About>

¹¹<http://dublincore.org/>

¹²<http://bibliontology.com/>

¹³<http://vocab.org/frbr/core.html>

¹⁴<http://www.umbel.org/vocabulary>

³<http://id.loc.gov/authorities/>

⁴<http://lobid.org/en/resource.html>

⁵http://www.d-nb.de/eng/hilfe/service/linked_data_service.htm

⁶<http://www.cs.vu.nl/STITCH/rameau/>

4.1 Example 1: The backlinking framework

The DnB node of figure 4 declares *skos:closeMatch* relationships between a number of its resources and the relevant ones of the LCSH node. As this is a result of manual annotation, LCSH is mostly unaware of this forward linking. There is a partial chance of interlinking discovery via the STITCH Rameau node, which acts as identity aligner. This chance is meager though, as only a fraction of LCSH resources (about 10%) are linked to STITCH Rameau entities.

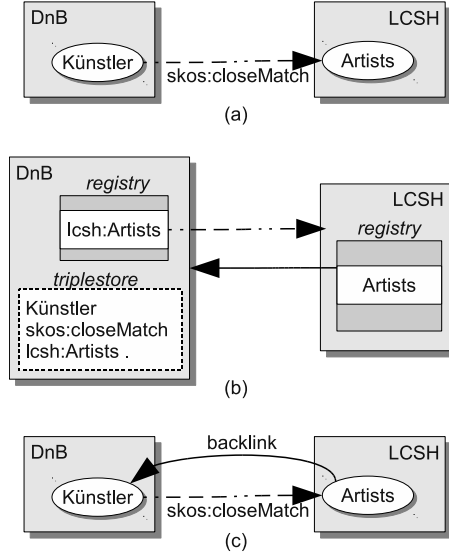


Figure 5: A backlinking example.

As an example, the entity “Künstler” is related in DnB node to the LCSH’s “Artists” entity with the following triple (actual entity identifiers and prefixes are replaced by their corresponding labels for clarity):

```
dnb:Künstler skos:closeMatch lcsch:Artists .
```

Conceptually, the triple creates a forward link between the graphs of the interrelated LOD repositories as shown in figure 5(a). When entering the above triple in the DnB repository with a tool aware of the backlinking framework, the following happen (figure 5(b)):

1. *lcsch:Artists* is entered in DnB’s registry, together with the access point of LCSH node (i.e. SPARQL endpoint).
2. The backlinking web service of the LCSH repository is notified via an HTTP request for the newly created reference. The HTTP request carries as parameters a) the referenced item *lcsch:Artists* and b) the access point of DnB (SPARQL endpoint).
3. If accepted, the request will lead to the addition in the LCSH’s registry of an entry keyed by *lcsch:Artists*, having as value the contact profile of DnB repository. Subsequently, the annotated registry can be queried in order to discover locations of external references to *lcsch:Artists*, creating in effect a backlink for *lcsch:Artists* pointing to the DnB node (figure 5(c)).

The mechanism described above makes the LCSH node aware of the fact that DnB references its “Artists” entity.

The action of LCSH node upon notification may vary, according to the degree of trust towards the DnB node:

- The reference can be fully accepted, leading LCSH to insert into its repository a symmetric triple containing a *skos:closeMatch* predicate towards “Künstler” DnB entity.
- The reference can be stored only in registry, for fast indexing or any other internal statistical processing. However it will not be added as a triple to the repository.
- Of course, the LCSH node could flatly reject the notification, ignoring the reference.

In any case, the reaction to the referencing notification is application-specific and the proposed backlinking framework is not involved into this handshaking process.

4.2 Example 2: Enabling navigation via backlinking

As the LOD-cloud expands and interconnects in higher degree, smart applications will appear with the ability to combine data from different LOD repositories and present a unified solution to the user. This data stitching can be achieved through URI-dereferencing crawling or distributed SPARQL queries. Either way, the backlinking framework presented herein can be used to support the process of distributed data gathering.

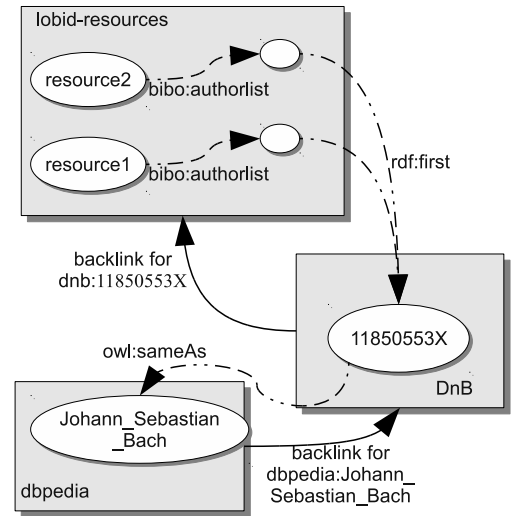


Figure 6: Discovery paths via backlinks.

As an example, consider the conceptual graph of figure 6 between three repositories containing information about the composer Johann Sebastian Bach. Arrows in dashed style indicate existing links between repositories, created when a repository uses a foreign URI in its triples. Solid line arrows indicate backlinking paths provided by the proposed framework. Clearly, when dbpedia is used as a starting point for navigation or querying, there is no way to arrive to lobid-resources repository without backlinking. For example, starting the information seeking process from dbpedia, only backlinks can provide a path for discovering entities

in lobid-resources repository (the *resource1*, *2* nodes of figure 6) that are related to the *dbpedia:Johann_Sebastian_Bach* entity. This is achieved by following the backlink for *dbpedia:Johann_Sebastian_Bach* and then the backlink for *dnb:11850553X*.

As argued before, the backlinking framework is an application-agnostic component. As such, the LOD repositories and querying applications must decide themselves in what degree do they trust the results fetched from remote nodes.

4.3 Example 3: Facilitating LOD maintenance

Peripheral repositories in figure 4 such as STITCH Rameau and VIAF are in reality interconnection hubs, gathering, aligning and redistributing entity identities of various LOD repositories. The creation and maintenance of such nodes involves parsing of RDF dumps and manual or semi-automatic alignment of entities. This process must be repeated in regular intervals, in order to synchronize the data on new entity additions or alterations. The backlinking framework provides an automatic way of adding equivalence or similarity statements, when a remote repository notifies the interconnection hub for the creation of an inter-relating triple involving a hub's resource.

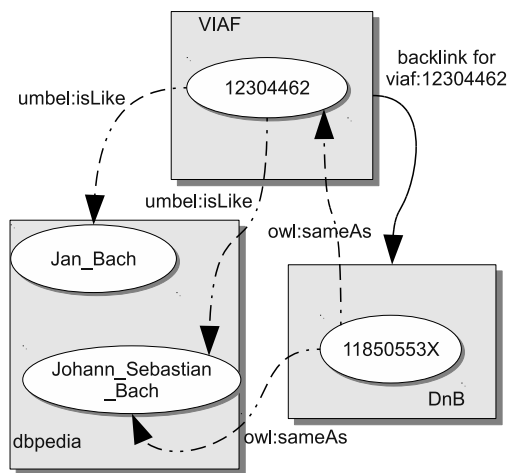


Figure 7: Checking identities via backlinked nodes.

In figure 7, another possible use of backlinking is demonstrated. The VIAF repository has two possible mappings in dbpedia for its defined entity *viaf:12304462*, representing the composer Johann Sebastian Bach. From these two mappings, only *dbpedia:Johann_Sebastian_Bach* is the correct one, while *dbpedia:Jan_Bach* refers to another person. This erroneous linking situation can be fixed: DnB vouches that the VIAF entity under question is the same as *dbpedia:Johann_Sebastian_Bach*. The VIAF repository can discover this fact only through backlinking to DnB. Should backlinking to DnB node be trusted, such a discovery could lead automatically the VIAF repository to increase its trust to the correct dbpedia mapping.

5. CONCLUSIONS AND FUTURE WORK

In this paper, a novel approach for providing richer connectivity between LOD repositories is presented. The provided connectivity is facilitated through the introduction of

a framework capable of aiding local repositories efficiently managing the references that are made to their defined resources from remote repositories. Such references function as backlinks.

The proposed framework does not impose the addition of triples within each participating LOD repository. Moreover, it can be employed from applications wishing to aggregate data from various LOD repositories through any kind of interlinking predicates. These features underpin the scalability and adaptivity of the proposed framework within an ever-growing environment such as the WoD.

Future work is targeted towards the implementation of the proposed framework and the subsequent design of applications that will demonstrate its usefulness.

6. REFERENCES

- [1] R. Isele, J. Umbrich, C. Bizer, and A. Harth. Ldspider: An open-source crawling framework for the web of linked data. In *9th International Semantic Web Conference (ISWC2010)*, November 2010.
- [2] G. Kobilarov, T. Scott, Y. Raimond, S. Oliver, C. Sizemore, M. Smethurst, C. Bizer, and R. Lee. Media meets semantic web — how the bbc uses dbpedia and linked data to make connections. In *Proceedings of the 6th European Semantic Web Conference on The Semantic Web: Research and Applications, ESWC 2009 Heraklion*, pages 723–737, Berlin, Heidelberg, 2009. Springer-Verlag.
- [3] A. Langegger, W. Wöß, and M. Blöchl. A semantic web middleware for virtual data integration on the web. In S. Bechhofer, M. Hauswirth, J. Hoffmann, and M. Koubarakis, editors, *ESWC*, volume 5021 of *Lecture Notes in Computer Science*, pages 493–507. Springer, 2008.
- [4] T. Omitola, C. L. Koumenides, I. O. Popov, Y. Yang, M. Salvadores, M. Szomszor, T. Berners-Lee, N. Gibbins, W. Hall, schraefel, and N. R. Shadbolt. Put in your postcode, out comes the data: A case study. In *The Semantic Web: Research and Applications, 7th Extended Semantic Web Conference, ESWC 2010, Heraklion, Crete, Greece, May 30 - June 3, 2010, Proceedings, Part I*, pages 318–332, 2010.
- [5] S. Tramp, P. Frischmuth, T. Ermilov, and S. Auer. Weaving a Social Data Web with Semantic Pingback. In P. Cimiano and H. Pinto, editors, *Proceedings of the EKAW 2010 - Knowledge Engineering and Knowledge Management by the Masses; 11th October-15th October 2010 - Lisbon, Portugal*, volume 6317 of *Lecture Notes in Artificial Intelligence (LNAI)*, pages 135–149, Berlin / Heidelberg, October 2010. Springer.