Get! Mimetypes! Right!

Christian Chiarcos
Applied Computational Linguistics Lab, Goethe University Frankfurt, Germany

Abstract

This paper identifies three technical requirements – availability of data, sustainable hosting and resolvable URIs for hosted data – as minimal pre-conditions for Linguistic Linked Open Data technology to develop towards a mature technological ecosystem that third party applications can build upon. While a critical amount of data is available (and it continues to grow), there does not seem to exist a hosting solution that combines the prospects of long-term availability with an unrestricted capability to support resolvable URIs. In particular, data hosting services do currently not allow data to be declared as RDF content by means of their media type (mime type), so that the capability of clients to recognize formats and to resolve URIs on that basis is severely limited.

2012 ACM Subject Classification Applied computing → Format and notation; Applied computing → Document management and text processing; Software and its engineering → Interoperability

Keywords and phrases data hosting, mimetypes, resolvability, URIs, Linked Data foundations

Digital Object Identifier 10.4230/OASIcs.LDK.2021.5

Category Crazy New Idea

Linked (Open) Data is generally considered to be the prototypical technology to implement the "FAIR Guiding Principles for scientific data management and stewardship" [4], and for the specific domain of language resources, Linguistic Linked Open Data (LLOD) comes with the promise to facilitate the integration of linguistic information from and across distributed resources. In the more general context of web technology, this represents probably the most promising way to address challenges of multilinguality – and indeed, since the publication of the OntoLex-Lemon vocabulary in 2016,[1] this is specifically the area where technology and data are most mature, and LLOD is on the verge of becoming a mainstream technology.

The success of the LLOD vision, however, crucially depends on finding solutions for three elementary problems:[2]

(1) It is necessary to provide a critical amount of data in RDF and with links,
(2) it is necessary to provide sustainable hosting solutions so that future applications can rely on the availability of a resource,
(3) it is necessary to provide resolvable URIs for the data such that it can be addressed as Linked Data and used as such.

The community has gone a long way since the inception of the LLOD cloud in 2010 [3], and especially in the lexical domain, the first challenge is basically overcome. Several established sub-communities in the field now support LLOD technology, e.g., the WordNet community [1], and massive amounts of OntoLex-compliant lexical data are available, covering more than 400 language varieties with substantial dictionaries, e.g., [2].

[1] `https://www.w3.org/2016/05/ontolex/`

[2] These are not the only aspects where LLOD technology suffers from bottle-necks. Problems also exist when it comes to tooling, ease of use, the challenges to develop agreed-upon vocabularies to exploit possible synergies, etc. However, these are challenges on the user side, and they can be addressed if researchers, users, providers and engineers devote time and energy. The problems mentioned here are more elementary in that they are necessary to constitute a technical environment to publish, access and maintain previously created data. Without hope for arriving at such an ecosystem within a reasonable time frame, enthusiasm, time and energy will be invested in vain and quickly decay.

3rd Conference on Language, Data and Knowledge (LDK 2021).
Editors: Dagmar Gromann, Gilles Sérasset, Thierry Declerck, John P. McCrae, Jorge Gracia, Julia Bosque-Gil, Fernando Bobillo, and Barbara Heinisch; Article No. 5; pp. 5:1–5:4
OpenAccess Series in Informatics
OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

As for the second challenge, the publication of data and its maintenance for subsequent use, replication and verification has been a problem for academic research in general. This is mostly caused by the fact that data is often produced in the context of temporary investments, e.g., as part of thesis projects or research grants. Traditionally, neither addressed questions of long-term data storage: A student will not have the resources and simply move on to other challenges after accomplishing a degree, and a fixed-term research project will eventually run out of funding. Publication via web sites may work for some time, but as soon as the local IT department or the hosting institution undergoes any form of major restructuring, much data is likely to be relocated – if not lost. So, unless designated efforts for preservation and link updates are being made, the life expectancy of a legacy dataset published in this way is at maybe, around 5 years after the project finished. Libraries may help here, but then, policies with respect to data hosting differ greatly, publications will always take priority in this context, much more so than data hosting, and resources are severely limited, e.g., in size of data dumps permitted.

Luckily, things improved greatly in this respect. With platforms such as Zenodo,[3] researchers now have the possibility to deposit their data under a persistent URL, with the large number of CLARIN centers established in the last decade,[4] there are regional solutions specifically for tools and data from natural language processing and the language sciences over all the EU, and with the growth of DARIAH[5] and SSHOC,[6] comparable ecosystems also emerge in the Humanities and Social Sciences. But even independently from designated research funding, established commercial solutions do exist which may depend to a lesser degree from central funding, e.g., GitHub[7] is occasionally used for the purpose. So, the challenge of data hosting has also been largely overcome.

At the moment, a major bottle-neck for LLOD technology is the third challenge. Open source RDF data will only be able to become Linked Open Data if the individual data points ("things") can be addressed with resolvable URIs. Looking at Zenodo as an example, it is possible to deposit RDF data, of course. And if that data uses persistent URIs created by a redirection service such as W3ID or Purl, it is possible to redirect them to the specific URL/DOI generated by Zenodo. So, they could resolve, in theory.

The problem here is that they can resolve only if the data is recognized as RDF data by an application that accesses the data dump. The standard way for doing so would be by an RDF mimetype such as text/turtle (etc., for other RDF formats). Unfortunately, the mimetype of data in Zenodo is either application/octet-stream or text/plain.

This means that applications need to guess the format if they attempt to resolve URIs against a resource. This can work, but it is unreliable. In particular, it will fail if URIs do not include the file ending (as recommended, because we have content negotiation, except not here), or if the data URI carries any flags after the file ending (e.g., "...?download=1").

Let's take the Jena-based service under `http://www.sparql.org` as an example, with a short query against `https://zenodo.org/record/4444132/files/crmtex.owl?download=1`:

```
SELECT *
WHERE { ?a ?b ?c } LIMIT 10
```

[3] https://zenodo.org/
[4] https://www.clarin.eu/
[5] https://www.dariah.eu/
[6] https://www.sshopencloud.eu/
[7] https://github.com/

The service of sparql.org does allow to query RDF data on the web, without the need to set up a local SPARQL end point or to download any data, so this is a nice demonstrator for RDF-based web services. Moreover, the SPARQL query can be added to the URI, so, it can be re-purposed in other web services and, for example, consulted via the LOAD keyword from a local SPARQL end point. With minimal effort, this web service is capable to demonstrate the key benefits of federation and information integration without putting the burden to maintain or set up any infrastructure on the developer of a particular query.[8]

Unfortunately, this fails with the original Zenodo data link.[9] In this case, it will work if flags are stripped and the file extension is recognized,[10] but this not robust (it is guesswork specific to this particular implementation and not guaranteed to work with other consumers). In essence, while the SPARQL query is portable and due to the use of W3C standards, the data is, as well, the behavior of your local triple store is somewhat unpredictable. Depending on specific heuristics to determine the content of the RDF data, it will perform differently (if at all).

The problem is not limited to the FROM keyword: With your local triple store, you might want to use the LOAD keyword of SPARQL, for example, to retrieve a remote data set. But again, the same problem arises if the mediatype of the data to be loaded from a remote host is not declared. Furthermore, the problem is not specific to Zenodo, it is only an example. In fact, I am not aware of any provider of LOD-compliant hosting services for an unrestricted pool of data providers. To illustrate a real-world example involving a commercial provider, GitHub displays its “raw” data similarly as text/plain. For example, the persistent URL http://purl.org/acoli/conll redirects to https://raw.githubusercontent.com/acoli-repo/conll-rdf/master/owl/conll.ttl, but this is exposed as text/plain, not text/turtle. Whether or not a particular SPARQL engine will be able to resolve this URI (note that – in accordance with LOD best practices –, the persistent URI does not include the file extension!) will vary across different implementations, giving the entire technology the appeal to be fragile and unreliable.

Fixing this by supporting RDF-compliant media types could unleash a wave of new demonstrators of the technology, that illustrate data re-use and integration from Zenodo and other portals. As it stands, these demonstrators often run against unstable university pages – or just quietly break. Having them run against data dumps hosted at Zenodo or other academic data maintainers would guarantee the necessary longevity to reliably demonstrate federated search to students, scholars and future generations.

Indeed, complementing existing hosting services with LOD-compliant, resolvable URIs would establish the minimal technical level of interoperability required to make existing (L)LOD data and services stable, sustainable and eventually operational. Moreover, reliable long-term hosting would enable commercial use cases. At the moment, the lack of confidence in long-term availability of LOD data sets represents a bias for the development of applications and services that depend on any such data. But only if Linked Data also works in a business context (and the potential is great), its vision and prospects will be able to unfold.

[8] In comparison to a local triple store there is a limitation in performance and scalability. But it is still an ideal, almost effortless environment for testing and demonstration.

[9] The following URI contains the corresponding query and the FROM clause points to the respective data source. The URI should resolve against a dynamically created query result. http://www.sparql.org/sparql?query=SELECT+*FROM+<https://zenodo.org/record/4444132/files/crmtex.owl?download=1>WHERE+{+?a+?b+?c+}+LIMIT+10&default-graph-uri=&output=xml&stylesheet=/xml-to-html.xsl.

[10] http://www.sparql.org/sparql?query=SELECT+*FROM+<https://zenodo.org/record/4444132/files/crmtex.owl>WHERE+{+?a+?b+?c+}+LIMIT+10&default-graph-uri=&output=xml&stylesheet=/xml-to-html.xsl

Overall, this is very easy to fix, and here comes a Crazy New Idea: Make a coordinated effort as a community to get providers of language resource infrastructures to support Linked Data compliant media types, e.g., petition repeatedly and massively to maintainers and developers of such infrastructures that data is declarable as text/turtle (etc.) than just text/plain or application/binary. After all, their decision to not support LOD-compliant mediatypes is a deliberate one, and it's not resulting from ignorance, but from a (somewhat lazy) risk-gain calculation: Data provided by a hosting service can be used to infuse malicious code into applications of clients, especially if it is automatically executed in the browser, and minimizing the number of supported mediatypes reduces this risk for the host, or better, it transfers the responsibility for executing malicious code from the host to the client. Given the current state of affairs, it is up to the providers and users of (Linguistic) Linked (Open) Data to explore that risk and to convince infrastructure providers that this risk is minimal (text/turtle is not interpreted by browsers, these days), that there is a potential gain for them (more functionalities, more popularity) and that there is a concrete need in their user community. Given the continued – and rising – popularity of Linguistic Linked Open Data, this point can be easily made, and – with the Cost Action Nexus Linguarum and several large-scale European and national projects based on this technology at this time – more easily so for language resources than for Linked Data in general.

It would be an exaggeration to call the idea to implement established standards crazy or even particularly innovative, but there is a new aspect I would like to throw into the discussion, that is, to address this technical problem also at the political level: Let's *collectively* approach infrastructure providers.

References

1 Francis Bond, Piek Vossen, John McCrae, and Christiane Fellbaum. CILI: The Collaborative InterLingual Index. In *Proc. of the 8th Global Wordnet Conference (GWC 2016)*, Bucharest, Romania, 2016.

2 Christian Chiarcos, Christian Fäth, and Maxim Ionov. The acoli dictionary graph. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 3281–3290, 2020.

3 Christian Chiarcos, John McCrae, Philipp Cimiano, and Christiane Fellbaum. Towards open data for linguistics: Linguistic linked data. In Alessandro Oltramari, Piek Vossen, Lu Qin, and Eduard Hovy, editors, *New Trends of Research in Ontologies and Lexical Resources: Ideas, Projects, Systems*, pages 7–25. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013. doi:10.1007/978-3-642-31782-8_2.

4 Mark D. Wilkinson, Michel Dumontier, IJsbrand Jan Aalbersberg, Gabrielle Appleton, Myles Axton, Arie Baak, Niklas Blomberg, Jan-Willem Boiten, Luiz Bonino da Silva Santos, Philip E. Bourne, Jildau Bouwman, Anthony J. Brookes, Tim Clark, Mercé Crosas, Ingrid Dillo, Olivier Dumon, Scott Edmunds, Chris T. Evelo, Richard Finkers, Alejandra Gonzalez-Beltran, Alasdair J.G. Gray, Paul Groth, Carole Goble, Jeffrey S. Grethe, Jaap Heringa, Peter A.C 't Hoen, Rob Hooft, Tobias Kuhn, Ruben Kok, Joost Kok, Scott J. Lusher, Maryann E. Martone, Albert Mons, Abel L. Packer, Bengt Persson, Philippe Rocca-Serra, Marco Roos, Rene van Schaik, Susanna-Assunta Sansone, Erik Schultes, Thierry Sengstag, Ted Slater, George Strawn, Morris A. Swertz, Mark Thompson, Johan van der Lei, Erik van Mulligen, Jan Velterop, Andra Waagmeester, Peter Wittenburg, Katherine Wolstencroft, Jun Zhao, and Barend Mons. The FAIR Guiding Principles for scientific data management and stewardship. *Scientific Data*, 3(1), 2016. doi:10.1038/sdata.2016.18.