Yuzu: Publishing Any Data as Linked Data

John P. McCrae

Insight Centre for Data Analytics, National University of Ireland, Galway
john@mccr.ae

Abstract. Linked data is one of the most important methods for improving the applicability of data, however most data is not in linked data formats and raising it to linked data is still a significant challenge. We present Yuzu, an application that makes it easy to host legacy data in JSON, XML or CSV as linked data, while providing a clean interface with advanced features. The ease-of-use of this framework is shown by its adoption for a number of existing datasets including WordNet.

Keywords: linked data, data frontend, data conversion

1 Introduction

Linked data [1] has been identified as one of the major ways to present data for knowledge discovery and has been shown to improve the quality and the usefulness of datasets [12]. However, a major challenge remains the conversion of datasets into linked data [5, 10]. This is frequently caused by the fact that data is in legacy formats such as CSV, XML or JSON and the conversion from these formats into RDF often represents much of the effort of a project. In recent years, a number of efforts have been made to make RDF and linked data work with these formats in particular, CSV on the Web [15] and JSON-LD [13], and these formats should lower the barrier to entry to users of linked data.

In this paper, we present the Yuzu platform[1], a frontend for linked data, like Pubby [4] or LodLive [3]. This platform can, in contrast to existing systems, aims to be free from strict restrictions about the format of the data, instead assuming that the data can be understood even in legacy formats with a small amount of metadata. This system also removes the need to run a separate SPARQL database and instead allows simple SPARQL access to data with some limitations 'out-of-the-box'. In addition, this platform implements many features that are required to make data easy-to-work with including content negotiation and automatic backups based on hashes [7].

2 Handling data in legacy formats

Data can be structured in three main ways: firstly tabular data which is serialized by means of table format where data is separated typically by a tab or comma.

[1] Yuzu is available at https://github.com/jmccrae/yuzu

Secondly, hierarchical data is structured in a flat tree and XML and JSON are the two most popular serialization methods. Finally, graph structured data has the most freedom in its representation, and RDF is the most commonly found form of this data, but databases based on this model can have a significant performance gap, which is called the "RDF tax" [2]. The Yuzu model is to keep documents in the format that is created but enable querying over them as if they were graph-based linked data. All conversions are provided using existing standards such wherever possible. and as such the input to Yuzu is the dataset as a single ZIP file containing all the data files in a some mix of XML, CSV and JSON.

2.1 JSON-LD and XML

JSON documents in Yuzu can be understood by means of a context document and it is required that each data either is a JSON-LD document with a `@context` element or that in the containing folder there is a `context.json`, which is used for indexing and is returned with the `Link` header [13, §6.8]. XML is mapped also using the JSON-LD context file and we assume a simple generic mapping method, whereby attributes and subtags (if there is no text context) are treated as name/value pairs in an object. If this is not possible the `@value` special property is used. Alternatively, a mapping may be provided using the LIXR mapping language [9].

2.2 CSV

CSV conversion is based on the CSV on the Web standard's recipe for creating RDF data [14], which we implement as part of the Yuzu model. Generation of RDF data from this CSV is provided in *standard mode* such that extra data for querying is available to the user. Each CSV file is described by means of an extra metadata file in the form of the Tabular Data Metadata Vocabulary [11], which is in fact another JSON-LD file. In the case where no mapping is found a default empty tabular metadata file is created and used to map the CSV into RDF. In the interface, data that was originally in CSV is presented to the user in a tabular form, however the RDF data can be obtained by means of content negotiation.

3 Cheap, robust SPARQL querying

SPARQL provides a powerful and effective method for querying data on the Web, however it provides significant challenges for hosts wishing to provide fast access with limited resources. SPARQL is a very free query language and it is easy to devise queries that are very hard to answer, and even worse this can easily be caused by typos[2].

[2] e.g., a typo in a variable name will not be detected in SPARQL and will turn a query that could be answered with an inner join to a query that can only be answered with the more expensive cross join

We employ a pre-processor that attempts to find a fixed subset of documents that have a given property and then creating a mini-dataset to evaluate the query on. This means certain queries, for example those which rely on `FILTER` constraints to do most of the document selection, will not be possible to execute, but more typical queries, such as documents with a list of properties can be more readily executed. We believe that this provides a good perfomance balance and will continue to evaluate this balance in our deployed instances. Of course, a full SPARQL endpoint can be used along with our this method to support all SPARQL queries.

4 Hashing, permalinks and backups

A major issue that faces data users is that data frequently becomes unavailable or has changed in a manner that makes it difficult to reuse. In order to combat this, Yuzu takes a hash of the overall dataset and a hash of each individual file in the dataset. The hash of each individual file can be used to look up any individual resource.

Secondly, each Yuzu instance may allocate a certain amount of space to back up parts of other resources. This back-up procedure is implemented by a method based on the Kademlia [6] protocol. Each Yuzu instance generates at start-up a unique identifier and checks the identifier of each of its peers (from a fixed list of peers). Then the files in the dataset are posted to each of the peers and the peers store those files that are closest in the XOR distance between the file's hash and the instance's hash, up to the limit of files that are there for back-up. Then when resolving a 'permalink', if the hash does not correspond to any of the file in this resources dataset the system redirects to another host, whose instance hash is closer to the requested hash.

5 Conclusions and current deployments

The ease-of-use of the Yuzu system has been deployed to host a number of datasets: originally it was developed for the WordNet dataset[3], and this is still a supported theme of the system. Since then, Yuzu has been applied to large datasets, such as Linghub [8], and has been used to host a large number of smaller datasets. The robust theming and stability of the interface has allowed datasets to be hosted even on very low-resourced virtual machines, even while allowing querying using SPARQL.

Acknowledgements

This research was supported by the Science Foundation Ireland under Grant Number SFI/12/RC/2289 (Insight)

[3] http://wordnet-rdf.princeton.edu

References

1. Bizer, C., Heath, T., Berners-Lee, T.: Linked data-the story so far. Semantic Services, Interoperability and Web Applications: Emerging Concepts pp. 205–227 (2009)
2. Boncz, P., Erling, O., Pham, M.D.: Advances in large-scale RDF data management. In: Linked Open Data–Creating Knowledge Out of Interlinked Data, pp. 21–44. Springer (2014)
3. Camarda, D.V., Mazzini, S., Antonuccio, A.: LodLive, exploring the web of data. In: Proceedings of the 8th International Conference on Semantic Systems. pp. 197–200. ACM (2012)
4. Cyganiak, R., Bizer, C.: Pubby-a linked data frontend for sparql endpoints (2008), http://www4.wiwiss.fu-berlin.de/pubby/
5. Ehrmann, M., Ceconi, F., Vannella, D., McCrae, J.P., Cimiano, P., Navigli, R.: Representing Multilingual Data as Linked Data: the Case of BabelNet 2.0. In: Proceedings of the 9th Language Resource and Evaluation Conference. pp. 401–408 (2014)
6. Maymounkov, P., Mazieres, D.: Kademlia: A peer-to-peer information system based on the xor metric. In: Peer-to-Peer Systems, pp. 53–65. Springer (2002)
7. McCrae, J.P., Bordea, G., Buitelaar, P.: Linked data and text mining as an enabler for reproducible research. In: Proceedings of the Workshop on Cross-Platform Text Mining and Natural Language Processing Interoperability (2016)
8. McCrae, J.P., Cimiano, P.: Linghub: a linked data based portal supporting the discovery of language resources. In: Joint Proceedings of the Posters and Demos Track of 11th International Conference on Semantic Systems-SEMANTiCS 2015 and 1st Workshop on Data Science: Methods, Technology and Applications (DSci15). pp. 88–91 (2015)
9. McCrae, J.P., Cimiano, P.: LIXR: Quick, succinct conversion of XML to RDF and back again. In: Proceedings of the ISWC 2016 Posters and Demo Track (2016)
10. O'Riain, S., Curry, E., Harth, A.: XBRL and open data for global financial ecosystems: A linked data approach. International Journal of Accounting Information Systems 13(2), 141–162 (2012)
11. Pollock, R., Tennison, J., Kellogg, G., Herman, I.: Metadata vocabulary for tabular data. W3C recommendation, World Wide Web Consortium (2015)
12. Schultz, A., Matteini, A., Isele, R., Mendes, P.N., Bizer, C., Becker, C.: Ldif-a framework for large-scale linked data integration. In: 21st International World Wide Web Conference (WWW 2012), Developers Track, Lyon, France (2012)
13. Sporny, M., Longley, D., Kellogg, G., Lanthaler, M., Lindstrm, N.: Json-ld 1.0. W3C recommendation, World Wide Web Consortium (2014)
14. Tandy, J., Herman, I., Kellogg, G.: Generating RDF from tabular data on the web. W3C recommendation, World Wide Web Consortium (2015)
15. Tennison, J., Kellogg, G., Herman, I.: Model for tabular data and metadata on the web. W3C recommendation, World Wide Web Consortium (2015)