LOD-a-lot
A Queryable Dump of the LOD Cloud

Javier D. Fernández[1], Wouter Beek[2],
Miguel A. Martínez-Prieto[3], and Mario Arias[4]

[1] Vienna University of Economics and Business, AU
[2] Dept. of Computer Science, VU University Amsterdam, NL
[3] Dept. of Computer Science, Universidad de Valladolid, SP
[4] Mario Arias Software, UK

javier.fernandez@wu.ac.at, w.g.j.beek@vu.nl,
migumar2@infor.uva.es, mario.arias@gmail.com

Abstract. LOD-a-lot democratizes the access to the Linked Open Data (LOD) Cloud by serving more than 28 billion unique triples from 650K datasets over a single self-indexed file. This corpus can be queried online with a sustainable Linked Data Fragments interface, or downloaded and consumed locally: LOD-a-lot is easy to deploy and demands affordable resources (524 GB of disk space and 15.7 GB of RAM), enabling Web-scale repeatable experimentation and research even by standard laptops.

1 Introduction

The last decade has seen an impressive growth of the Linked Open Data (LOD) community, which promotes to use the Resource Description Framework (RDF) [21] to publicly share semi-structured data on the Web and to connect different data items by reusing HTTP International Resource Identifiers (IRIs) across data sources [4]. Besides HTTP access to RDF data, publishers also provide RDF dataset dumps (for download), and query endpoints that expose various capabilities, ranging from basic queries in RESTful APIs, such as Linked Data Fragments (LDF) [23], to SQL-like structured queries using SPARQL [10].

Although the LOD paradigm should provide access to a huge distributed knowledge base that can be browsed and queried online, efficient web-scale consumption of LOD remains a tedious issue in practice. Consider, for example, retrieving all entities with the label *"Tim Berners-Lee"*, which could be easily formulated as a SPARQL query: `select distinct ?x { ?x owl:sameAs*/rdfs:label "Tim Berners-Lee" }`. Given the distributed nature of LOD, the resolution of this simple query would require either of the following approaches:

- *Download, index and query datasets from the LOD Cloud locally*, which will result in scalability issues, and it is impractical for simple query resolution.
- *Run a federated query against all known sources* [6], which is as good as the query endpoints that it relies on. Unfortunately, SPARQL endpoints have low availability [8,22], and federated queries are difficult to optimize beyond a limited number of sources [18].

– *Browse online sources in a "follow-your-nose" way* [12], i.e., traversing the universally distributed graph of shared knowledge on-the-fly. In practice, many IRIs do not dereference, and since our query contains no IRI (only an RDF literal), it is not even clear where graph traversal should start (or end).

Thus, all three strategies present drawbacks, making it unfeasible to perform an evaluation of a simple query on the Web [13]. Some of these issues are partially solved by services like Datahub[5] and LOD Laundromat[6] [2], which provide central catalogs for discovering and accessing cached versions of LOD datasets. However, data consumers still need to navigate and process large corpora, consisting of thousands of dumps or endpoints, to conduct large-scale experiments.

In this paper, we propose the **LOD-a-lot dataset** as a means to offer low-cost consumption of a large portion of LOD. We integrate 650K datasets from the LOD Cloud (crawled and cleaned by LOD Laundromat [2]) into a single self-indexed HDT [9] file. The resultant HDT is conveniently small and can be directly queried by data consumers with a limited memory footprint, promoting experiments at web scale with commodity hardware. LOD-a-lot contains 28 billion unique triples and, to the best of our knowledge, is the first approach to provide *an indexed and ready-to-consume crawl of a large portion of the LOD Cloud*. In addition, we serve an LDF interface to LOD-a-lot as an online, central and sustainable way to serve structured querying of LOD.

The paper is organized as follows. Section 2 presents LOD-a-lot and its main benefits. Section 3 describes the available interfaces and tools to work with LOD-a-lot. We summarize LOD-a-lot statistics in Section 4, and describe potential use cases for it in Section 5. Section 6 concludes and devises future work.

2 LOD-a-lot: Concepts and Benefits

LOD-a-lot proposes an effective way of packaging a standards compliant subset of the LOD Cloud into a ready-to-use file comprising data from LOD Laundromat.

LOD Laundromat [2] is a service that crawls, cleans and republishes LOD datasets from Datahub, as well as other seeds manually collected. As illustrated in Figure 1, each dataset is first cleaned to improve data quality: i) syntax errors are detected and heuristics are used to recover from them (when possible); ii) duplicated RDF statements are removed; iii) *Skolemization* is performed to replace blank nodes with well-known IRIs[7]; and iv) the cleaned dataset is lexicographically sorted and *gzipped* using N-Triples [21]. The current version (May 2015) is composed of 657,902 datasets and contains more than 38 billion triples (including between-dataset duplicates). Finally, each dataset is serialized in HDT (for download), and is also published as an LDF [23] endpoint.

Header-Dictionary-Triples (HDT) [9] is a binary compression format and – at the same time – a self-contained and queryable data store for RDF. HDT

[5] See `https://datahub.io`
[6] See `http://lodlaundromat.org`
[7] See `https://www.w3.org/TR/rdf11-concepts/#section-skolemization`

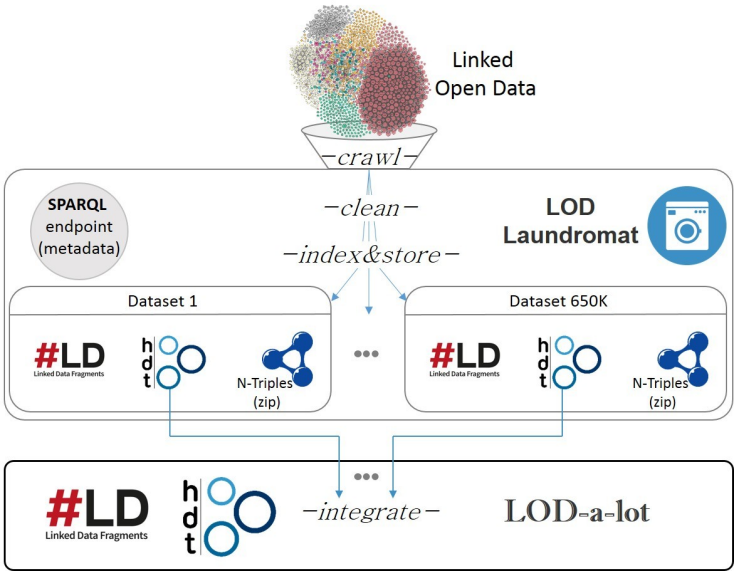

Fig. 1. LOD-a-lot overview and data flow.

represents its main components (Dictionary and Triples) with compact data structures that enable storing, parsing and loading Big Semantic Data in compressed space. HDT data are indexed by subject, and therefore can be used to efficiently resolve subject-bounded *Triple Pattern* (TP) queries[8] (as well as the unbounded query `???`) [9]. The so-called HDT-Focused on Querying (HDT-FoQ) [16] extends HDT with two indexes (enabling predicate and object-based access, respectively) than can be created by the HDT consumer in order to speed up all TP queries. Altogether, HDT can be used as a storage backend for large-scale graph data that achieves competitive querying performance [16].

Linked Data Fragments (LDF) [23] is aimed at improving the scalability and availability of SPARQL endpoints by minimizing server-side processing and moving intelligence to the client. LDF allows simple Triple Patterns to be queried, where results are retrieved incrementally through pagination. Each of these pages (referred to as *fragments*) includes the estimated results and hypermedia controls (using the Hydra Vocabulary [15]), such that clients can perform query planning, retrieve all fragments, and join sub-query results locally. As such, server load is minimized and large data collections can be exposed with high availability. Given that HDT provides fast, low-cost TP resolution, LDF has been traditionally used in combination with HDT.

In spite of the inherent benefits of LOD Laundromat to conduct large-scale experiments [3], consumers still need to access/query each single dataset/endpoint independently, which results in additional overheads when analyzing the full corpus as a whole. LOD-a-lot tackles this issue and provides a unified view of

[8] That is, `SPO`, `SP?`, `S?O` and `S??` patterns.

all the RDF data crawled, cleaned and indexed in LOD Laundromat. To do so, we carefully integrate all the 650K HDT datasets into a single HDT file. In order to improve the scalability of such process, we perform parallel and incrementally large merges of HDT datasets, integrating Dictionary and Triples components. In addition to the HDT file, we also create and expose the HDT-FoQ index, saving such resource-consuming task to the consumer[9]. Finally, we also serve all the integrated information through an LDF interface (see Figure 1).

The resultant LOD-a-lot dataset contains more than 28B triples (see Section 4 for further statistics) and is aimed at easy online and offline consumption and reuse. Specifically, LOD-a-lot has the following properties:

- **Standards-compliance.** The LOD Laundromat cleaning process and the HDT conversion guarantee that the indexed data is standards-compliant [2].
- **Volume & Variety.** LOD-a-lot consists of over 28 billion triples (one of the largest single RDF dataset) and merges more than 650K datasets, which cover a large subset of the topic domains in LOD.
- **Accessibility.** The combination of HDT and LDF in LOD-a-lot allows users to perform structured queries through a uniform access point that is standards-compliant and self-descriptive through Hydra [15].
- **Scalability & Availability.** Most LOD query endpoints are either exposing a small dataset, have low availability, or are too expensive to maintain [3]. LOD-a-lot alleviates these problems for online and offline data consumption: HDT is highly compressed and can resolve triple pattern queries at rest, with a very limited memory footprint (in practice, 3% of the total dataset size). In turn, LDF deploys such functionality online and minimizes the server burden, pushing the composition of more complex queries to the client.
- **Ease of (re)use.** Because LOD-a-lot is just one file, it can be downloaded, copied, or linked to easily.
- **Cost-effectiveness.** Due to the HDT compression technique, the hardware footprint of LOD-a-lot is relatively small, requiring 524 GB of (solid-state) disk space and (when queried) 15.7 GB of RAM. At the time of writing the combined cost of these two hardware resources is approximately 305 euros.

3 Availability and Sustainability

LOD-a-lot is available at `http://purl.org/HDT/lod-a-lot` and listed in the `datahub.io` catalog[10], where we provide the following access to the dataset:

- **HDT Dump**, including additional HDT-FoQ indexes to speed up query processing, released under the "creative commons by-sa 3.0" license[11].
- **LDF interface**, to serve online SPARQL resolution using LDF clients.
- **VoID[12] description** of the dataset to help in automatic discovery services.

[9] HDT creation took 64 h & 170GB RAM. HDT-FoQ took 8 h & 250GB RAM.
[10] See `https://datahub.io/dataset/lod-a-lot`
[11] See `https://creativecommons.org/licenses/by-sa/3.0/`
[12] See `https://www.w3.org/TR/void/`

#Triples	#Subjects	#Predicates	#Objects	#Common SO	#Literals
28,362,198,927	3,214,347,198	1,168,932	3,178,409,386	1,298,808,567	1,302,285,394

Table 1. LOD-a-lot summary statistics.

The sustainability of LOD-a-lot is supported by the join effort of the LOD Laundromat and HDT projects. These projects, together with LDF, have been running for the last 3-6 years and are well-established in the area. We also plan a regular update policy of LOD-a-lot (along with new LOD Laundromat crawls).

Finally, note that the LOD-a-lot file can be exploited using all available HDT tools, including C++ and Java libraries, easy deployment with Docker and integration with other open source projects such as Apache Jena and Tinkerpop.[13]

LOD-a-lot can be cited canonically as "*Fernández, J. D., Beek, W., Martínez-Prieto, M.A., and Arias, M. LOD-a-lot: A Queryable Dump of the LOD cloud (2017).* `http://purl.org/HDT/lod-a-lot`."

4 LOD-a-lot Statistics Summary

A simple analysis of LOD-a-lot reports some interesting statistics about its features. Table 1 first compares the number of unique triples, and different subjects, predicates, and objects used in our dataset. The two-rightmost columns also report the number of *common* subjects and objects, i.e. those terms playing both roles in the dataset, and the total number of literal objects. Results are in line with the widespread perception that the number of predicates is very limited w.r.t the number of triples (in this case, 1M predicates in 28B triples. i.e. less than 0.004%) due to vocabulary reuse. A more elaborated analysis (Figure 2, middle) shows that predicates follow a *power-law* distribution, where a long-tail of predicates is barely used while a limited set of predicates appears in a great number of triples.

Interestingly, almost the same number of subjects and objects (3B terms) are used in LOD-a-lot. The high proportion w.r.t the number of triples (11%) shows a low reuse of such terms. Figure 2 further elaborates on this and depicts subject (left) and object (right) distributions. Power-laws are reported in both cases, but a longer tail is drawn for objects with massive (up to 1B) repetitions. Finally, note two interesting numbers to understand the underlying dataset structure: i) around 40% of subjects and objects play both roles, which means that it is easy to find chain paths of, at least, two connected triples; and ii) more than 1.3B of objects are literals, so 41% of object nodes have no output links.

A *space complexity* analysis shows that the HDT LOD-a-lot dump encodes 28B triples in 304 GB: 133GB are used for compressing the Dictionary, and 171GB for the Triples component. HDT-FoQ indexes are also built to speed up all TP queries over the queryable dump: these additional structures use 220 GB.

Finally, we performed a deployment test (using the HDT-C++ library) on a modest computer[14], resulting in a load time of only 144 seconds and a memory

[13] See `https://github.com/rdfhdt`

[14] 8 cores (2.6 GHz), RAM 32 GB and a SATA HDD on Ubuntu 14.04.5 LTS.

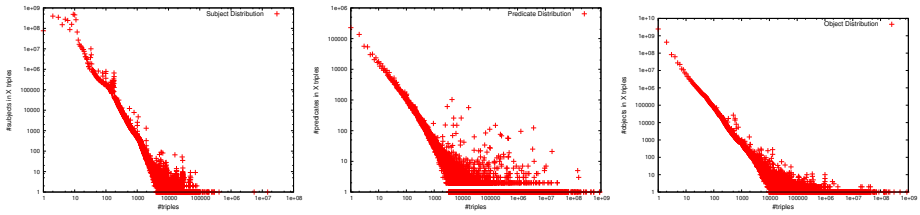

Fig. 2. Distribution of subjects, predicates, and objects in LOD-a-lot (log-log scale).

footprint of 15.7 GB of RAM (≈ 3% of the total dataset size). Furthermore, LDF queries (with 100 results as page size) are resolved at the level of milliseconds. This shows the LOD-a-lot affordable cost to manage and query 28B triples.

5 Relevance of the Dataset

This section describes three focused use cases for LOD-a-lot.

Query resolution at Web scale (UC1) is still an open challenge. Besides the aforementioned drawbacks of query federation [18,19]) and follow-your-nose traversal querying [13], centralized approaches such as Swoogle[15] and Sindice [20] are already discontinued. The OpenLink Software's LOD Cloud Cache[16] maintains a SPARQL endpoint of a portion of the LOD Cloud. However, it only reports 4B triples and has disruptions[17]. Furthermore, the system suffers from the traditional size/time restrictions of SPARQL endpoints and simple unbounded queries (e.g. getting labels of `owl:samAs` entities[18]) incurs in timeouts. LOD-a-lot promotes query resolution at Web scale not only by actually serving such service for the indexed 28B triples, but it also shows the feasibility, scalability and efficiency of a centralized approach based on HDT and LDF.

Evaluation and benchmarking (UC2) have gain increasingly attention in the Semantic Web community (see the Linked Data Benchmark Council project [5] for a general overview). However, Semantic Web evaluations lack of scale, variety and present a static or even synthetic dataset [3]. The Billion Triple Challenge (BTC) [14], the WebDataCommons microdata, RDFa and Microformat dataset series [17] assist in this context by crawling RDF data from the Web and providing a single integrated dataset. However, BTC is limited to 4B triples[19] and uses a minimum sample of each crawled data source, which provides an incomplete view of the data. In turn, the WebDataCommons dataset scales in size (44B triples[20]) but the focus is on microdata and thus its variety and general application is very limited in practice. LOD Laundromat addresses

[15] See `http://swoogle.umbc.edu/`
[16] See `http://lod.openlinksw.com/`
[17] See SPARQLES report: `https://goo.gl/1qJzDG`
[18] `SELECT * WHERE {?s owl:sameAs ?o. ?s rdfs:label ?x . ?o rdfs:label ?y}`
[19] See `http://km.aifb.kit.edu/projects/btc-2014/`
[20] See `http://webdatacommons.org/structureddata/2016-10/stats/stats.html`

this issue and republishes heterogeneous RDF datasets, but these have to be managed independently, which can result in a pain point for consumers. Thus, LOD-a-lot integrates the main advantages of all these proposals in terms of size (28B triples), variety (650K datasets) and single access point. LOD-a-lot is extremely easy and efficient to deploy in a local environment (via HDT), which allows Semantic Web academics and practitioners to run experiments over the largest and most heterogeneous, indexed and ready-to-consume RDF dataset.

RDF metrics and analytics (UC3) are widely adopted for SPARQL query optimization techniques [11] in order to find the most optimal query plan. However, few studies inspect structural properties of real-world RDF data at Web scale [7] and, even those, only involve few million triples. More recently, the potential of LOD Laundromat has been exploited to characterize the quality of the data [1]. LOD-a-lot characteristics (see Section 2) democratize the computation of RDF metrics and analytics at Web scale (see the degrees in Figure 2 as a practical example). Furthermore, particular metrics can take advantage of the HDT components in isolation, e.g. knowing the average length of URIs and literals would only scan the Dictionary (collecting all terms), whereas computing the in-degrees of object would only access the Triples (indexing the graph).

In addition, we also envision further practical applications for entity linking and data enrichment (e.g. leveraging in-links and `owl:sameAs` related entities), ranking of entities and vocabularies (e.g. analyzing their use), data summarization and other data mining techniques (e.g. finding commonalities in the data).

6 Conclusions and Future Work

The steady adoption of Linked Open Data (LOD) in recent years has led to a significant increase in the number and volume of RDF datasets. Today, problems such as data discovery and structured querying at web scale remain open challenges given the distributed nature of LOD.

This paper has presented LOD-a-lot, a simple and cost-effective way to query and study a large copy of the LOD Cloud. LOD-a-lot recollects all data gathered from the LOD Laundromat service and exposes a single HDT file, which can be queried online for free, and that can be downloaded locally and queried over commodity hardware. Requiring 524 GB of disk space and 15.7 GB of RAM, LOD-a-lot allows more than 28 billion unique triples to be queried using hardware costing – at the time of writing – 305 euro.

We plan to update LOD-a-lot regularly and include further datasets from the LOD Cloud. We are also working on a novel HDT variation to index quad information and thus keep track of the the input sources contributing to LOD-a-lot. Altogether, we expect LOD-a-lot to democratize the access to LOD and be one of the references for low-cost Web-scale evaluations.

References

1. Beek, W., Ilievski, F., Debattista, J., Schlobach, S., Wielemaker, J.: *Literally* Better: Analyzing and Improving the Quality of Literals. Semantic Web Journal (2017)

2. Beek, W., Rietveld, L., Bazoobandi, H.R., Wielemaker, J., Schlobach, S.: LOD Laundromat: A Uniform Way of Publishing other People's Dirty Data. In: Proc. of ISWC. pp. 213–228 (2014)
3. Beek, W., Rietveld, L., Ilievski, F., Schlobach, S.: LOD Lab: Scalable Linked Data processing. In: Reasoning Web: Logical Foundation of Knowledge Graph Construction and Query Answering, pp. 124–155. Springer International Publishing (2017)
4. Bizer, C., Heath, T., Berners-Lee, T.: Linked Data: The Story So Far. International Journal on Semantic Web and Information Systems 5(3), 1–22 (2009)
5. Boncz, P., Fundulaki, I., Gubichev, A., Larriba-Pey, J., Neumann, T.: The Linked Data Benchmark Council Project. Datenbank-Spektrum 13(2), 121–129 (2013)
6. Buil-Aranda, C., Arenas, M., Corcho, O., Polleres, A.: Federating Queries in SPARQL 1.1: Syntax, Semantics and Evaluation. JWS 18(1), 1–17 (2013)
7. Ding, L., Finin, T.: Characterizing the Semantic Web on the Web. In: Proc. of ISWC. pp. 242–257 (2006)
8. Ermilov, I., Lehmann, J., Martin, M., Auer, S.: LODStats: The Data Web Census Dataset. In: Proc. of ISWC. pp. 38–46 (2016)
9. Fernández, J.D., Martínez-Prieto, M.A., Gutiérrez, C., Polleres, A., Arias, M.: Binary RDF Representation for Publication and Exchange (HDT). JWS 19, 22–41 (2013)
10. Garlik, S.H., Seaborne, A., Prud'hommeaux, E.: SPARQL 1.1 Query Language, W3C Recommendation (2013), `https://www.w3.org/TR/sparql11-query/`
11. Gubichev, A., Neumann, T.: Exploiting the Query Structure for Efficient Join Ordering in SPARQL Queries. In: Proc. of EDBT. pp. 439–450 (2014)
12. Hartig, O.: SQUIN: a Traversal Based Query Execution System for the Web of Linked Data. In: Proc. of SIGMOD. pp. 1081–1084 (2013)
13. Hartig, O., Pirrò, G.: A Context-based Semantics for SPARQL Property Paths over the Web. In: Proc. of ESWC. pp. 71–87 (2015)
14. Käfer, T., Harth, A.: Billion Triples Challenge Data Set (2014), `http://km.aifb.kit.edu/projects/btc-2014/`
15. Lanthaler, M., Gütl, C.: Hydra: A Vocabulary for Hypermedia-Driven Web APIs. vol. CEUR-996 (2013)
16. Martínez-Prieto, M.A., Arias, M., Fernández, J.D.: Exchange and Consumption of Huge RDF Data. In: Proc. of ESWC. pp. 437–452 (2012)
17. Meusel, R., Petrovski, P., Bizer, C.: The WebDataCommons Microdata, RDFa and Microformat Dataset Series. In: Proc. of ISWC. pp. 277–292 (2014)
18. Millard, I.C., Glaser, H., Salvadores, M., Shadbolt, N.: Consuming Multiple Linked Data Sources: Challenges and Experiences. In: Proc. of COLD. vol. CEUR-665, pp. 37–48 (2010)
19. Oguz, D., Ergenc, B., Yin, S., Dikenelli, O., Hameurlain, A.: Federated Query Processing on Linked Data: a Qualitative Survey and Open Challenges. The Knowledge Engineering Review 30(5), 545–563 (2015)
20. Oren, E., Delbru, R., Catasta, M., Cyganiak, R., Stenzhorn, H., Tummarello, G.: Sindice.com: a Document-Oriented Lookup Index for Open Linked Data. International Journal of Metadata, Semantics and Ontologies 3(1), 37–52 (2008)
21. Schreiber, G., Raimond, Y.: RDF Primer 1.1., W3C Working Group Note (2014), `https://www.w3.org/TR/2014/NOTE-rdf11-primer-20140225/`
22. Vandenbussche, P.Y., Umbrich, J., Matteis, L., Hogan, A., Buil-Aranda, C.: SPARQLES: Monitoring Public SPARQL Endpoints. Semantic Web Journal (2017)
23. Verborgh, R., Vander Sande, M., Hartig, O., Van Herwegen, J., De Vocht, L., De Meester, B., Haesendonck, G., Colpaert, P.: Triple Pattern Fragments: a Low-cost Knowledge Graph Interface for the Web. JWS 37–38, 184–206 (2016)