Generating and Exploiting Semantically Enriched, Integrated, Linked and Open Museum Data

Sotirios Angelis and Konstantinos Kotis

Intelligent Systems Lab, Department of Cultural Technology and Communication, University of the Aegean, 81100 Mytilene, Greece
{sang, kotis}@aegean.gr

Abstract. The work presented in this paper is engaging with and contributes to the implementation and evaluation of Semantic Web applications in the cultural Linked Open Data (LOD) domain. The main goal is the semantic integration, enrichment and interlinking of data that are generated through the documentation process of artworks and cultural heritage objects. This is accomplished by using state-of-the-art technologies and current standards of the Semantic Web (RDF, OWL, SPARQL), as well as widely accepted models and vocabularies relevant to the cultural domain (Dublin Core, SKOS, Europeana Data Model). A set of specialized tools such as KARMA and OpenRefine/RDF-extension is being used and evaluated in order to achieve the semantic integration of museum data from heterogeneous sources. Interlinking is achieved using tools such as Silk and OpenRefine/RDF-extension, discovering links (at the back-end) between disparate datasets and other external data sources such as DBpedia and Wikidata that enrich the source data. Finally, a front-end Web application is developed in order to exploit the semantically integrated and enriched museum data, and further interlink (and enrich) them (at application run-time), with the data sources of DBpedia and Europeana. The paper discusses engineering choices made for the evaluation of the proposed framework/pipeline.

Keywords: Semantic data integration · Semantic enrichment · Museum LOD

1 Introduction

Museums and other cultural heritage organizations, while documenting artworks, artists and cultural heritage objects, are creating a great amount of data. In the past decade, there have been numerous attempts to take advantage of Semantic Web Technologies for linking and opening this (meta)data to the public as Linked Open Data (LOD). Based on the lessons learned from this experience, the work presented in this paper utilizes and evaluates related Semantic Web technologies with museum (meta)data, aiming at: a) integrating (combining the museum data stored in different sources to provide a unified view of them), enriching (adding relevant information to data records) and interlinking (finding similarities between datasets and connecting similar entities) datasets...
form multiple heterogeneous museum collections, b) developing an application that allows end-users (humans and machines) to exploit artworks and artists from an integrated and enriched dataset, and c) discussing related challenges and engineering choices made towards proposing a framework/pipeline for exploiting semantically integrated and enriched museum LOD.

Specifically, the stages followed in the proposed framework/pipeline are presented in Fig. 1, depicting two distinct parts, the “data-preparation and application back-end” part (orange rectangle) and the “application front-end” part (green rectangle).

Fig. 1. The proposed pipeline for semantic integration and exploitation of cultural museum data (Color figure online)

In order to semantically integrate multiple datasets from different and heterogeneous sources, resulting to a single source with unified information, the first part of the pipeline is responsible for a) the selection of a data model that sufficiently describes cultural heritage data, b) the RDFization process i.e., transforming data to the Resource Description Framework (RDF) format, c) the Link Discovery (LD) process, and d) the management of the semantic queries. The model used in this current work is the Europeana Data Model (EDM) [1], which is a widely-used model designed to describe the Europeana Collection [2]. RDFization is the process followed to convert the raw data from their original format (e.g., CSV, XML) into RDF and store them in a single RDF graph, semantically enriched with cultural metadata through the selected data model (ontology). LD is the process followed to discover similarities and create links between entities that are stored in the RDF graph and links to other external data sources that promote the enrichment of source data.

The second part of the framework/pipeline concerns the design and development of the front-end application that provides a user interface for exploiting information in the semantically integrated and enriched dataset. The front-end application is designed to be data-centric and drive the user to discover information and relations about artworks and artists by querying the interlinked museum data in a unified manner. It provides internal hyperlinks that present the details of each entity that is requested by the user (e.g., Pablo Picasso, The Starry Night), as well as external hyperlinks to DBpedia [3], Wikidata [4], and Europeana. The external links are created and stored during the LD process, and they can be updated. New links can be discovered in real-time by the application.
The main contribution of this work is to utilize and evaluate Semantic Technologies towards shaping an efficient framework/pipeline for exploiting semantically enriched, integrated, linked and open museum data. Furthermore, this work contributes a functional and reusable application for the exploitation of such data by human and software agents. This paper is structured as follows: Sect. 2 presents selective related work in cultural heritage semantic data integration and exploitation. Section 3 presents the datasets used for the evaluation of the proposed framework/pipeline, the RDFization process, the front-end Web application infrastructure, and its implementation. Section 4 presents the Semantic Technologies and tools used in the evaluation scenario and system. Section 5 discusses the engineering choices made for this work. Finally, Sect. 6 presents key achievements and future work.

2 Related Work

As the interest in Semantic Web Technologies for the cultural domain is growing, there have been several attempts and projects for converting cultural heritage raw data into LOD. We evaluate and compare our work with a representative set of ones that are relatively recent and closely related to ours.

In the work of Dragoni et al. [5], authors present the process of mapping the metadata from small size collections with little-known artists to the LOD cloud, exposing the created knowledge base by using LOD format, making it available to third-party services. A framework is proposed for a) the conversion of data to RDF with the use of a custom ontology that extends the Europeana Data Model, and b) the enrichment of metadata from content found on the Web. The metadata enrichment is then succeeded by linking to external knowledge bases such as DBpedia and with the use of Natural Language Processing (NLP) methods for information extraction from Web pages containing relevant details for the records.

In Szekely et al. [6], authors describe the process and the lessons learned in mapping the metadata of the Smithsonian American Art Museum’s (SAAM) objects to a custom SAAM ontology, that extends EDM with subclasses and sub-properties to represent attributes unique to SAAM. In this work the mapping is done by KARMA integration tool, while the linking and enrichment was focused to the interrelation of the museum’s artists with entities in DBpedia, Getty Union List of Artist Names and the Rijksmuseum dataset. The paper also discusses the challenges encountered in data preparation, mapping columns to classes, connecting the classes and mapping based on field values.

In the work of Stolpe et al. [7], the conversion of the Yellow List of architecturally and culturally valuable buildings in Oslo, from Excel spreadsheets to Linked Open Data, is presented. The data was mapped in the CIDOC Conceptual Reference Model [8] and converted by XLWrap4 package, a spreadsheet-to-RDF wrapper. The entities of the RDF graph were linked to DBpedia and exposed via a SPARQL endpoint based on a Joseki triple store. A Web application that plots the results of SPARQL queries onto a Google Map based on the coordinates in the Yellow List, was developed to help users to explore the data.

These related works start with raw cultural data and convert them to RDF. Their main goal is to expose the data of a single dataset as LOD, while the main goal of our work is to
semantically integrate, enrich and link disparate and heterogeneous cultural data in a way that it can be queried in a unified manner. Each of the related works have a single dataset to deal with, while two of them aim to achieve more detailed (enriched) descriptions of their data by engineering and using custom ontologies. Another difference is that, while dealing only with one dataset, related works do not need a LD process for interlinking between heterogeneous museum data, as needed in our case.

We also acknowledge the related work of Amsterdam Museum Linked Open Data [9] and The REACH Project [10], which present well-defined methodologies, the first on generating Linked Open museum Data and the latter on semantically integrating cultural data and exposing them via a web application. Each of them covers in detail parts of our proposed pipeline while working with data from one source (in contrast to our work).

3 The Approach

To evaluate the proposed framework/pipeline for exploiting semantically enriched, integrated, linked and open museum data, we designed and implemented a prototype system. This system is engineered in two distinct parts (or sub-systems), the back-end one that supports the data preparation and the application infrastructure, and the front-end one that supports the functionality of the Web application for querying and presenting the interlinked and enriched museum data in a unified and open manner (for human and software agents to exploit).

3.1 Back-End

The main goal here is to convert cultural data into RDF format so they can be easily integrated and linked. The datasets that were chosen for evaluation purposes are published as research datasets by two well-known museums, the Museum of Modern Art and the Carnegie Museum of Art.

**MoMA Collection.** The collection of the Museum of Modern Art – MoMA [11], includes almost 200,000 artworks from all over the world. This collection extents in the field of visual arts, such as paintings, sculptures, photographs, architectural designs etc. In order to contribute and help in the understanding of its collection, MoMA created a research dataset. The dataset contains 138,000 records, representing all the works that have been accessioned into MoMA’s collection and cataloged in its’ database. It contains basic metadata for each artwork, including title, artist, date of creation, medium etc. The dataset is available in CSV and JSON format and it is published in the public domain using a CC0 License.

**Carnegie Collection.** The Carnegie Museum of Art – CMoA has published a dataset [12] that contains information and metadata for almost every artwork from the museum’s collection and counts over 28,000 artworks. The dataset includes artworks from all the departments of the museum, that divides to fine arts, photography, modern art and architecture. The dataset is available in CSV and JSON format and it is published under CC0 License.
**RDFization Process.** In order to reduce the heterogeneity of the available evaluation data and to combine the datasets in one unified RDF graph, the selection of the characteristics for the conceptual model for semantically enriching the RDF data with cultural metadata was based on two assumptions/limitations: a) to be present in both of the datasets and b) to match a semantic type that is described in the EDM. Additionally, following directions of the EDM and common principles/guidelines for the LOD, new classes and URLs were created, based on the unique identifiers of every record in the datasets. The properties that were mapped to semantic types for each collection are the following: Title, Date, Medium, ObjectID, Classification, URL, ThumbnailURL, ConstituentID, Artist, ArtistBio, BeginDate, EndDate, AggregationID, Aggregation_uri, Artist_uri, Artwork_uri. The resulted, after the mapping, model is shaped in Fig. 2.

![Fig. 2. The proposed Data Model](image)

After the model was defined, the next step in the process was the conversion of data into RDF format. This was accomplished with the selection (among others) and use of the semantic integration tools KARMA [13] and OpenRefine [14]. Using both these tools we created the RDF graph based on the model and populated it with RDF data generated from each collection’s dataset. The generated graphs were both stored in the same RDF triple store, forming a single integrated graph.

**Links Discovery Process.** The LOD standard [15] indicates firstly the conversion of data in a machine-readable format through an expressive model and consequently the discovery of relative information and the creation of external links to it. The linking tools that were chosen (among others) for evaluating the discovery of relative information to the data were the SILK Framework [16] and the OpenRefine Reconciliation Service [17]. The linking tasks that created were targeted at the model’s main classes, i.e., Artwork and Artist. The first linking attempts were aiming to discover semantic similarities between entities of the two graphs, stored in the local triple store. Subsequently, the LD tasks searched and created links to the knowledge graphs of DBpedia [3] and Wikidata [4]. The similarity metrics were equality or string-matching metrics (Levenshtein distance), comparing a) for Artists: the attributes name, birth date, death date, and nationality, and b) for Artworks: the attributes title, creation date and creator.
3.2 Front-End

The developed front-end Web application is a type of single page application and it is compatible with modern desktop and mobile browsers. The application’s infrastructure, as depicted in the overall systems’ architectural design of Fig. 3, includes: a) the storage of the integrated and interlinked cultural datasets and their external connections to the Knowledge Graphs of DBpedia and Wikidata, b) a server that receives the requests and allows the interaction with the stored data, at the back-end and c) the Graphical User Interface (GUI) with the embedded functionality at the front-end.

The whole functionality of the front-end application is developed at the client-side and, although it is connected to a Fuseki server (for the proof-of-concept system evaluation), it can be functionally connected to any SPARQL endpoint that exposes the integrated datasets and supports federated queries. The chosen triple store for the evaluation system is the TDB2 [18] which is a part of the Apache Jena platform [19] and is configured to be integrated with the Fuseki2 server [20]. The evaluated graph that is stored at the TDB and includes the linked data (RDF data that created in the RDFization process for all artworks and artists from MoMA and CMoA collections process, plus the sameAs triples that generated during the LD process) from the museum collections alongside the external connections (sameAs triples to DBpedia and Wikidata that discovered in LD process), contains more than 1,645,000 triples.

Front-end Web Application Implementation. The front-end Web application was developed with the use of two mainstream front-end frameworks: React.js and Bootstrap 4. The design is following a reusable component-based approach, in which interdependent blocks are developed and they compose to build the UI of the application. Therefore, each component was developed to satisfy the needs of each distinct functionality of the application.

The front-end Web application is a template-driven search form that allows users to create semantics-based queries about artists and their artworks. The values of each field of the search form are gathered and synthesized in such a way that a SPARQL query is formed, which is then submitted for execution by the Fuseki Server. Each field
is shaping a form that is configured properly to implement different criteria based on given attributes. During the form submission, a part of the final SPARQL query is built automatically.

To make search easier, text suggestions with possible matched entries are automatically filling the text fields, while users are typing, based on the entities that are present in the datasets. This is accomplished each time the user types a character in the search field, by forming and submitting a query for checking its match against the semantic types of the stored graph.

The query results are received from the server in JSON format and converted to tabular and triples format, as well as presented graphically as a directed graph. The graphical representation uses the D3.js library and a modified version of D3RDF\(^1\) tool to create a directed graph based on the SPARQL query results.

The application creates pages dynamically in order to present the details of each resource and to perform additional linking tasks, discovering connections with external sources. The additional linking process searches for same resources in Europeana and DBpedia graphs by creating a federated SPARQL query based on the semantic type of the resource and the attribute that describes it. If results are returned, they are filtered based on chosen similarity metric (Jaro-distance), and a \textit{sameAs} triple is added in the graph (if not present already). This feature is currently implemented only for the Artist entities.

**Front-end Application Use Case Scenario.** For demonstration reasons, let assume the user searches information about “Portraits painted by Andy Warhol”. The generated SPARQL query for this search is presented below:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX edm: <http://www.europeana.eu/schemas/edm/>
PREFIX dc: <http://purl.org/dc/elements/1.1/>  
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
PREFIX dct: <http://purl.org/dc/terms/>
SELECT distinct * WHERE{
    ?cho dc:title ?title.
    ?cho dc:creator ?artist.
    ?artist skos:prefLabel ?name.
    ?cho dc:type ?classification.
FILTER regex(?title, "Portrait", "i")
FILTER regex(?name, "Andy Warhol", "i")
FILTER regex(?classification, "Painting", "i")}
```

The results that are returned from the server for the SPARQL query (Fig. 4) contain the cultural heritage object’s URI, title and classification, the artist’s URI and name, displayed in table format. The results demonstrate that users are able to search the museum data in a unified manner, as the information about the artworks are retrieved from both CMoA and MoMA collections. The details of the resource returned when the user selects the URI of the artist are presented in Fig. 5. Apart from the information about

\(^1\) [https://github.com/Rathachai/d3rdf].
the artist, the details present the discovered external links, in this case link to DBpedia and to Wikidata.

![Diagram](image)

**Fig. 4.** Search results for query “Portraits painted by Andy Warhol”

![Diagram](image)

**Fig. 5.** Details of the resource “Andy Warhol”

## 4 Framework and System Evaluation

The Semantic Web technologies and tools used for the implementation of the framework’s evaluation system can be organized in three types: a) RDFization tools, b) Link Discovery tools, c) Infrastructure and Storage technologies. All the experiments ran on Ubuntu 18.04 laptop, with i7 2nd Gen CPU and 6 GB of RAM.
4.1 RDFization

**EDM.** Europeana Data Model (EDM) was designed as a tool for collection, connection and enrichment of the metadata that describe the objects offered to Europeana Collection. In order to avoid management of a collection that every content provider uses different description metadata, EDM uses a well-defined set of elements which includes classes and attributes from commonly used vocabularies as well as some that were developed for this specific reason.

**KARMA.** KARMA [13] is an open-source tool for data integration. It has features to process and semantically integrate data provided by different sources, such as databases, Web APIs, or XML, CSV, JSON files.

Data processing was performed with use of expressions written in Python, called PyTransforms. They were applied to the data for cleaning and the creation of new columns with the URIs based on the identifiers of the records in order to be mapped to the classes that are defined accordingly to EDM. Consequently, with the use of KARMA’s graphical interface, the data were mapped to their semantic types. The last step was to export the data in RDF format. The conversion of MoMA dataset (~128,000 records) form CSV to RDF lasted 1 min and 55 s, while the conversion of CmoA dataset (~28,000 records) from JSON to RDF, lasted 50 s.

**OpenRefine.** OpenRefine [14] is a platform for data exploration, cleaning and linking. It offers a variety of features for exploration, classification, and transformation of data. The data processing tasks run General Refine Expression Language – GREL, which comes embedded to OpenRefine. RDF-extension provides tools and features to OpenRefine to convert data into RDF format. Specifically, it adds a graphical interface and allows the user to import ontologies and vocabularies and to map the data with their attributes.

The data were cleaned, and new URI columns were created with use of GREL expressions. Afterwards, the model that the RDF graph will be based on, was created in the GUI of RDF-extension. Every potential node of the graph was mapped to the values of a column and assigned to a value type, such as text, number, date, URI etc. The conversion of MoMA dataset (~128,000 records) form CSV to RDF lasted 1 min and 45 s, while the conversion of CMoA dataset (~28,000 records) from JSON to RDF, lasted 35 s.

**Other Tools.** We have also experimented with RDF-gen [21] and DataBearing [22] tools that use their custom SPARQL-like domain-specific languages for data integration and RDFization process. Their main advantages are the quick implementation of the data model for common types and the good performance on big datasets. However, due to incomplete results obtained based on lack of resources, we decided not to include them in this paper.

4.2 Link Discovery Process

**OpenRefine.** One feature that is provided in OpenRefine platform is the Reconciliation Service. This feature allows LD between different data sources and it can work either with
RDF files or with SPARQL endpoints. The user must define the entity type, the resource class and optionally attributes that will provide higher precision at the discovered links.

The linking process between Artists from MoMA and CMoA collections lasted about 2 h and the results were 28,184 links for the whole datasets, from which the 640 were unique. The linking process between Artists from MoMA collection and DBpedia endpoint did not return any results, while the one between Artists from MoMA collection and Wikidata endpoint lasted about 12 h and the results were 82,185 links for the whole dataset, from which the 6,800 were unique.

**SILK.** Silk [16] is an open-source platform for data integration of heterogeneous sources. Silk is using a declarative language for the definition of the linking type and the criteria based on which the discovered entities will be matched successfully to the resources. Silk uses SPARQL protocol to send queries to local and external endpoints. The LD process runs linking tasks that are created with use of the declarative language or the graphical interface. Every Linking tasks defines the necessary data transformations and the similarity metrics that are used to discover relation between two entities.

The linking process between Artists from MoMA and CMoA collections lasted about 12 s and the results were 1,042 links. The linking process between Artists from MoMA collection and DBpedia endpoint lasted about 35 min and the results were 760 links. The linking process between Artists from CMoA collection and DBpedia endpoint lasted about 7 min and 25 s and the results were 452 links.

Example triples stored in the RDF graph, after LD process are presented below:

```xml
<http://localhost:3000/moma/artist/4609> a <http://www.w3.org/2000/01/rdf-schema#Resource>, owl:Thing, edm:Agent;
    edm:begin "1881";
    edm:end "1973";
    owl:sameAs <http://dbpedia.org/resource/Pablo_Picasso>,
              <https://www.wikidata.org/wiki/Q5593>,
              <http://localhost:3000/moma/artist/4609>,
              <http://localhost:3000/cmoa/artist/b7eaca81-9a3a-404a-866e-05844d616971>;
    skos:note "(Spanish, 1881–1973)";
    skos:prefLabel "Pablo Picasso".

<http://localhost:3000/moma/76281/artwork> a edm:ProvidedCHO;
    dc:creator <http://localhost:3000/moma/artist/5471>;
    dc:date "1905";
    dc:title "Fifth Avenue Critics";
    dc:type "Print";
    dct:medium "Etching";
    ore:isAggregatedBy <http://localhost:3000/moma/76281>;
    owl:sameAs <http://localhost:3000/cmoa/776ffb7d-e37b-4dce-8dcd-745786a17d95/artwork>.
```
4.3 Infrastructure and Data Storage

Fuseki2 [20] is a server that accepts SPARQL queries and is part of Apache Jena platform. It can be used as service, Web application or standalone server. Fuseki2 offers an environment for server management and it is designed to cooperate with TDB2 [18] triple-store to provide a reliable and functional storage level. Fuseki2 also supports SOH service (SPARQL over HTTP), in order to accept and execute SPARQL queries from HTTP requests. The design of the developed application was based on this feature. TDB2 is a part of the Apache Jena platform and is configured to integrate with the Fuseki2 server for storing RDF graphs.

5 Discussion

In order to propose a specific pipeline that fulfills the purpose of this research, several decisions had to be made about the choices of tools and technologies used in each stage. As a lesson learned, it must be stated that every stage of this pipeline is independent, and the choice made in each stage could be altered without significant impact on the performance of the overall process.

First of all, the datasets selected were in different formats (CSV and JSON) and were relatively big, containing together more than 150,000 records. This decision was made in order to be able to check integration and scalability issues and discover limitations on the approach. One limitation that we faced was during the attempt to apply semantic reasoning on the integrated data to infer new relations between the data and make the query results more efficient. Due to lack of computational power, chosen tool inefficiency and data model design issues, the inferencing process with the integrated reasoning services of Fuseki could not be completed for our RDF graphs.

The EDM is proposed for the semantic enriching because it is a widely-used and well-known model, it is a combination of commonly used vocabularies, it was designed for integration of heterogeneous museum data, and lastly because the integrated RDF data would be compatible with the Europeana Collection.

The tools were selected based on the criteria of efficiency, usability, cost, and reliability, and the majority of those that were evaluated were free, non-commercial, open-source and widely used. OpenRefine is proposed for the RDFization process, for its ease of use, mistakes avoidance design, usability on data manipulation and efficiency on data conversion from original data format to RDF. On the other hand, OpenRefine lacks on performance and choices during the LD process, so the proposal here is the Silk framework for its efficiency and variety of option that provides. Another lesson learned from limitations was related to the linking tasks that we’ve created on Silk: while were successfully discovered links between the museum datasets and DBpedia, they could not complete against the Europeana Collection. That was one of the reasons (as a workaround) that we’ve designed the front-end Web application instead to perform those LD tasks in a customized way.

Finally, Apache Jena platform is proposed for the infrastructure (RDF triple-store and SPARQL server) because it is one of the commonly used open-source and free Semantic Web frameworks, supports SPARQL queries over HTTP, and integrates TDB which is a high-performance triple store.
The source code of the developed application along with the integrated RDF data can be found on https://github.com/sotirisAng/search-on-cultural-lod.

6 Conclusion

This paper presented our contribution for an efficient framework/pipeline for semantically integrating, enriching, and linking open museum data as well as for exploiting such data (by end-users and by other applications) using a functional and reusable Web application. As presented in Fig. 6 (updated version of Fig. 1), the aim of the presented work is to propose a specific technological pipeline for efficient semantic integration and enrichment of museum data that can be queried either from humans or machines. Our future plans on this research line include:

- publish the data online (public URIs)
- expand the data model/ontology used for wider coverage of the metadata provided by the museum collections,
- experiment with the integration of more datasets in formats other than JSON and CSV (e.g., data in relational databases or in XML formats),
- update the linking tasks by adding more sophisticated similarity metrics, to achieve better results on the linking process
- expand the functionality of the Web application to perform linking tasks to more entities, not only between Artists.

Fig. 6. Selected technologies for the proposed framework/pipeline

References

1. Europeana: Definition of the Europeana Data Model (2017)
2. Europeana Collections. https://www.europeana.eu/portal/en. Accessed 11 Feb 2020
3. DBpedia, https://wiki.dbpedia.org/, last accessed 2020/08/23
4. Wikidata. https://www.wikidata.org/wiki/Wikidata:Main_Page. Accessed 23 Aug 2020
5. Dragoni, M., Cabrio, E., Tonelli, S., Villata, S.: Enriching a small artwork collection through semantic linking. In: Lecture Notes in Computer Science, pp. 724–740. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-319-34129-3_44
6. Szekely, P., et al.: Connecting the smithsonian american art museum to the linked data cloud. Lect. Notes Comput. Sci. 7882, 593–607 (2013). https://doi.org/10.1007/978-3-642-38288-8-40
7. Skjæveland, M.G.: From Spreadsheets to 5-star Linked Data in the Cultural Heritage Domain: A Case Study of the Yellow List Audun Stolpe. (2011)
8. CIDOC CRM. http://www.cidoc-crm.org/. Accessed 29 Sept 2020
9. De Boer, V., et al.: Amsterdam museum linked open data. Semant. Web. 4, 237–243 (2013). https://doi.org/10.3233/SW-2012-0074
10. Doulaverakis, C., Kompatsiaris, Y., Strintzis, M.G.: Ontology-based access to multimedia cultural heritage collections - the REACH project. In: EUROCON 2005 - International Conference Computer as a Tool. I, pp. 151–154 (2005). https://doi.org/10.1109/eurcon.2005.1629881
11. The Museum of Modern Art (MoMA) collection data. https://github.com/MuseumofModernArt/collection. Accessed 11 Feb 2020
12. The collection data of the Carnegie Museum of Art in Pittsburgh, Pennsylvania. https://github.com/cmoa/collection. Accessed 11 Feb 2020
13. Knoblock, C.A., et al.: Semi-automatically mapping structured sources into the semantic web. Lect. Notes Comput. Sci. 7295, 375–390 (2012). https://doi.org/10.1007/978-3-642-30284-8_32
14. OpenRefine, https://openrefine.org/. Accessed 11 Feb 2020
15. Linked Data - Design Issues. https://www.w3.org/DesignIssues/LinkedData. Accessed 23 Aug 2020
16. Volz, J., Bizer, C., Gaedke, M., Kobilarov, G.: Silk - a link discovery framework for the web of data. In: CEUR Workshop Proceedings, p. 538 (2009)
17. Reconciliation Service. https://github.com/OpenRefine/OpenRefine/wiki/Reconciliation-Service-API. Accessed 26 Aug 2020
18. TDB2. https://jena.apache.org/documentation/tdb2/. Accessed 26 Aug 2020
19. Apache Jena https://jena.apache.org/getting_started/index.html. Accessed 12 Feb 2020
20. Apache Jena Fuseki. https://jena.apache.org/documentation/fuseki2/. Accessed 12 Feb 2020
21. Santipantakis, G.M., Vouros, G.A., Kotis, K.I., Doulkeridis, C.: Rdf-gen: Generating RDF from streaming and archival data. ACM Int. Conf. Proc. Ser. (2018). https://doi.org/10.1145/3227609.3227658
22. DataBearings. https://sites.google.com/site/databearings/home. Accessed 03 Oct 2020