Providing Effective Visualizations over Big Linked Data

Short Research Paper

Federico Desimoni
federico.desimoni@unimore.it
"Enzo Ferrari" Engineering Department
University of Modena and Reggio Emilia
Modena, Italy

Laura Po
laura.po@unimore.it
"Enzo Ferrari" Engineering Department
University of Modena and Reggio Emilia
Modena, Italy

ABSTRACT

The number and the size of Linked Data sources are constantly increasing. In some lucky case, the data source is equipped with a tool that guides and helps the user during the exploration of the data, but in most cases, the data are published as an RDF dump through a SPARQL endpoint that can be accessed only through SPARQL queries. Although the RDF format was designed to be processed by machines, there is a strong need for visualization and exploration tools. Data visualizations make big and small linked data easier for the human brain to understand, and visualization also makes it easier to detect patterns, trends, and outliers in groups of data.

For this reason, we developed a tool called H-BOLD (High-level Visualization over Big Linked Open Data). H-BOLD aims to help the user exploring the content of a Linked Data by providing a high-level view of the structure of the dataset and an interactive exploration that allows users to focus on the connections and attributes of one or more classes. Moreover, it provides a visual interface for querying the endpoint that automatically generates SPARQL queries.

KEYWORDS

Linked Data, Visualization, Big Data, SPARQL, Semantic Web, Visual Querying, Data Visualization

1 INTRODUCTION

Since 2006, the year in which sir Tim Berners-Lee coined the term Linked Data (LD), leading to a new way in which data can be structured and accessed through the Internet, the number of LD exploded. Starting from governments, many institutions, enterprises, and privates adopted this method for publishing data. As a result, there is a huge number of LD that can be accessed through the Internet. Unfortunately, endpoints are big containers of triples. They can contain every kind of information and the high number of triples required to express a concept made LD visualization a non-trivial task. Due to the volume and the variety of information, it is hard to find a common procedure for exploring every dataset but nonetheless, several research groups tried to address this task [10, 12, 16, 17]. A hierarchical visualization can address the problem of information overloading, offering an effective mechanism for information abstraction and summarization. Additionally, an interactive exploration allows the user to understand step-by-step the content of even complex and big LD.

In this paper, we present a new version of the tool H-BOLD (High-level visualizations on Big Open Linked Data). H-BOLD enables the exploratory search and multilevel analysis of Big

LD by offering different levels of abstraction. The first version of H-BOLD appeared in 2018 [13, 15]. In this paper, we want to describe the re-engineering process that has been carried out and the additional features developed.

The architecture of the tool is introduced in Section 2, while the new features implemented are reported in Section 3. Related work are described in Section 4. Section 5 sketches conclusion and future work.

2 H-BOLD

H-BOLD[1] (High-level visualizations on Big Open Linked Data) is a tool, available online, for visualizing, and interacting with LD. It is defined in the context of hierarchical and interactive visual exploration and analysis over LD [13].

H-BOLD starts from our past experience with the tool LODeX, a tool able to automatically provide a summarization of a LD, including its inferred schema [1–3, 5], and tried to overcome the main limitations arose during its evaluation [4]. The main goal of H-BOLD was to facilitate the exploration of LD with a high number of classes. The current architecture of H-BOLD is shown in Figure 1 where the updates, with respect the previous version [13], are highlighted in blue.

H-BOLD is composed of a server layer and a presentation layer that will be described in the following.

2.1 Server layer

The server layer aims to generate high-level representations of a set of LD starting from a list of SPARQL endpoints. The outputs are the Cluster Schema (a high level representation of a complex and big source) and the Schema Summary (a low level representation of the instanced classes within a source). The SPARQL endpoint list is created starting from the old list of endpoints used in LODeX and adding new SPARQL endpoints that are present on Open data portals. Moreover, we enable users to manually add a new URL for a SPARQL endpoint they wish to visualize and explore in H-BOLD. On each endpoint a set of queries is executed in order to extract structural and statistical information that describe the LD, this phase is called Index Extraction. In particular, the indexes are the number of instances, the number of classes, the list of classes with the respective properties and the number of instances belonging to a specific class. The Index Extraction is able to deal with the performance issues of the different implementations of SPARQL endpoints by using *pattern strategies* [1].

Starting from the indexes is then possible to create a *Schema Summary* of the LD, a pseudograph that represents, through nodes and arches, the relations between the various instantiated classes of the dataset [2, 5]. The Schema Summary is a good approach to compactly represent a RDF dataset, however when

---

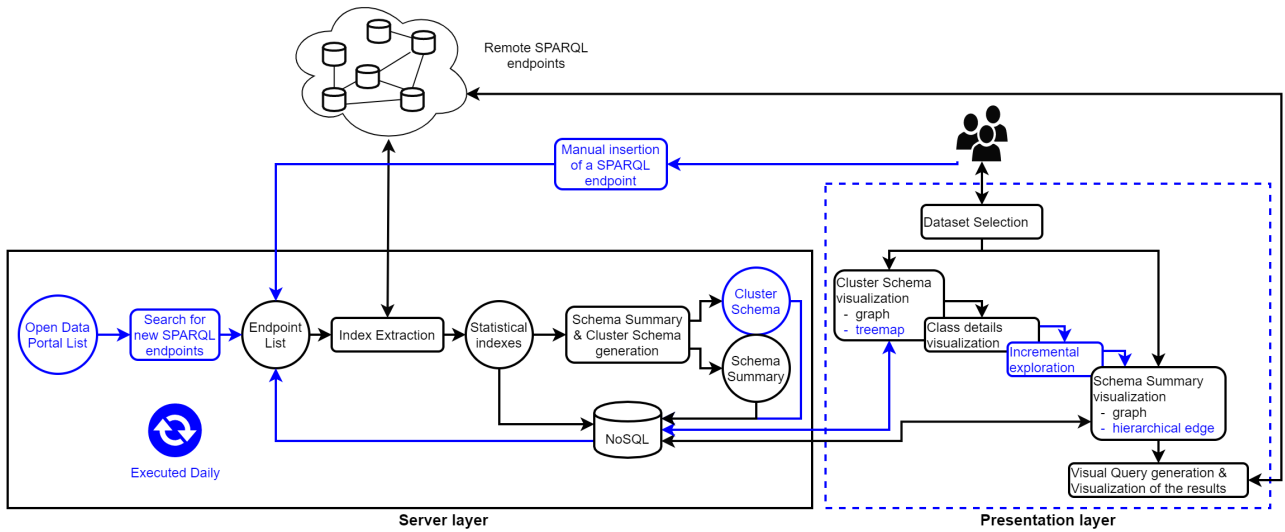[1]https://dbgroup.ing.unimore.it/hbold/

Figure 1: Architecture and Workflow of H-BOLD

we are dealing with big sources, it happens that the number of classes is high, thus the graph contains a high number of nodes and the visualization results complex and confused. On the Schema Summary, a set of community detection techniques has been used to create a high-level visualization for Big LD [15]. The classes, of the Schema Summary, are grouped into Clusters, therefore, a Cluster Schema is generated for each LD where nodes are groups of classes and arches are connections among these Clusters. In the clustering of the Schema Summary, the possibility that a node belongs to several Clusters is avoided. The labels in the Cluster Schema are assigned based on the degree (the sum of in-degree and out-degree) of the classes (nodes) that are represented by the cluster. For the formal definition of the Schema Summary and Cluster Schema see [15].

The Schema Summary and Cluster Schema offer several advantages: they can be easily memorized and retrieved on the MongoDB improving data recovery performance and graph visualization.

2.2 Presentation layer

In the presentation layer, the first step for a user is the selection of a dataset, then, the user can start the exploration of the Cluster Schema or the exploration of the Schema Summary. The first is more concise, while the second is the complete visualization of the structural information on the LD.

If the user opts for the Cluster Schema (see Figure 2, steps 1), he/she will see a shrank representation of the Schema Summary that is obtained by applying a community detection algorithm. From the Cluster Schema, by selecting a class within a cluster, a new visualization focused on the selected class is proposed to the user. The user might then further explore the class, its connections with other classes and its attributes, or can iteratively increase the graph displayed by expanding the connections starting from some classes (nodes in the graph). In each partial representation of the Schema Summary, the user is informed about the percentage of the instances represented by the graph and the total number of nodes (see Figure 2, steps 2 and 3). This expansion can be repeated until all the classes are displayed, as in the Schema Summary visualization.

Another option for the user to start is from the exploration of the Schema Summary, here, he/she will see a complete graph containing all the instantiated classes of the LD (see Figure 2, step 4). The user can focus on a particular class and explore its attributes and properties. Figure 2 shows the visualization steps in H-BOLD starting from the Cluster Schema and then selecting a class and expanding the graph till the visualization of the entire Schema Summary over the Scholarly LD[2].

3 RE-ENGINEERING THE TOOL

The H-BOLD application has been renovated both in the server layer and in the presentation layer. The server layer is implemented entirely in python and it uses MongoDB for the storage of information. With respect to the previous version of H-BOLD, it has been equipped with new features for making the application more responsive to the user requests. First of all, to enhance the understanding of the behavior of the tool, a guideline has been drawn up: the application code has been studied and various flowcharts have been extracted. A process for continuously updating and enriching our collection of SPARQL endpoints has been implemented (see Subsection 3.1 for further details). The Cluster Schema creation has been re-implemented and included in the Server Layer, while previously was calculated on-the-fly in the Presentation Layer (Subsection 3.2). The list of SPARQL endpoints has been enriched thanks to the integration of three new open data portals (Subsection 3.3).

The presentation layer went through a complete re-engineerization. Due to many incompatibilities between the current browsers and the version of Polymer[3] in use (i.e. a javascript framework developed by Google that helps building web application through the concept of web components), we were forced to re-implement the presentation layer of H-BOLD.

We started searching alternatives that could replace Polymer. At the end of this research, we decided to move towards Bootstrap[4], a javascript framework developed by Twitter for building and styling web applications. Therefore, the interface of the web application has been completed rebuilt with Bootstrap. The

---

[2]http://www.scholarlydata.org/

[3]https://www.polymer-project.org/

[4]https://getbootstrap.com/
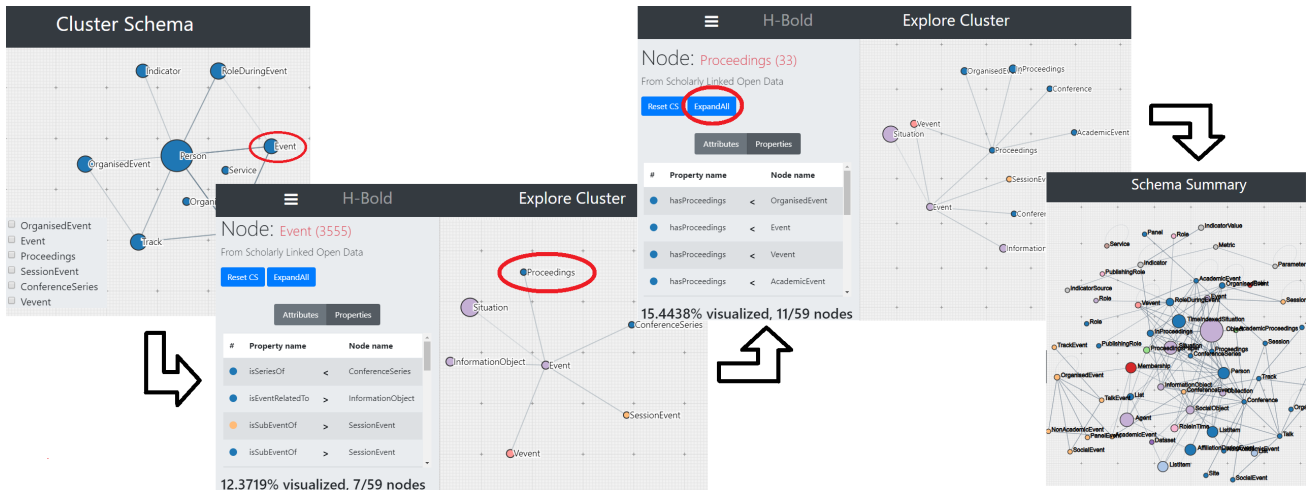
Figure 2: Step-by-step visualization of the Scholary LD. From left to right: 1) visualization of the Cluster Schema, 2) exploration of the "Event" class, 3) further expansion and exploration of the dataset, 4) complete visualization of the Schema Summary.

javascript graphic library, D3[5], already in use in H-BOLD, has been integrated with Bootstrap to create interactive visualizations. Moreover, new features ameliorate H-BOLD. A new interface for allowing user to manually insert a SPARQL endpoint has been added (Subsection 3.4). New visualization layouts have been conceived for the exploration of the Cluster Schema and the Schema Summary (Subsection 3.5).

3.1 Endpoint extraction automation and updates

Adding new content inside a SPARQL endpoint is a simple task since the only constraint is that data must be represented in one of the RDF serialization formats. Adding new classes or relations is just as simple as adding new instances. As a consequence, the structure and also the content of a LD could change very often. In H-BOLD, we want to display the most updated version of the dataset we indexed. For this reason, we automated the procedure of indexes extraction and Schema Summary and Cluster Schema generation to run daily. Working with SPARQL Endpoints since 2014, we noticed two important aspects. First, a SPARQL Endpoint might be often not available[6], but this does not mean that it is completely out of order, it might work again after 1 or 2 days. Second, LD do not change daily, they usually changed weekly, or monthly, or do not change ever. For these reasons, it is useless to run the index extraction over all the datasets daily, it is enough to run it weekly. However, since a SPARQL endpoint might not be available one day, and maybe be online the next day, we should check its availability. Therefore, we decided to store the date of the last index extraction for each SPARQL endpoint. If the last index extraction was executed more than seven days before, then we do not update the information for that LD, however, if some things went wrong with the last index extraction because the endpoint was not available, we can repeat the index extraction every day.

3.2 Cluster Schema visualization and generation

The Cluster Schema is the core part that has been inserted in H-BOLD, starting from the predecessor LODeX. The Cluster Schema allows to visualize in an aggregated way all the classes of a LD. In the previous demo of H-BOLD [13], the Cluster Schema was calculated on-the-fly by running the community detection algorithm each time a user asked to see a Cluster Schema. This procedure shows different weak points. Since the Cluster Schema is computed over the Schema Summary, if the Schema Summary does not change then the Cluster Schema will not change neither, so it does not make sense to recompute the Cluster Schema on each user click. Moreover, even if the community detection algorithm took short time to run, the user had to wait the information to be both transformed and loaded before its visualization. For these reasons, we inserted the computation of the community detection algorithm on the server side layer and the storage of the Cluster Schema in the MongoDB. Now, the Cluster Schema is computed only once, after the index extraction procedure and the Schema Summary computation, and then stored in the DB. Therefore, both the Schema Summary and Cluster Schema can be visualized by directly querying the DB. Experimental results showed that, on half of the SPARQL endpoints stored in H-BOLD, the time needed to display the Cluster Schema to the user is decreased by the 35%.

3.3 Automatic insertion of SPARQL endpoints by crawling open data portals

In the previous version of H-BOLD, the list of datasets that users could visualize was manually created from a list of SPARQL endpoints available on DataHub. Unfortunately, the endpoint we queried for obtaining such information is no more available but the same data can be found at this link [7]. Moreover, that list was quite old and some of the endpoints were no longer available. For this reason, we conducted a research to identify which were the most relevant open data portals that contains links to SPARQL endpoints. Then, we extended both the server

---
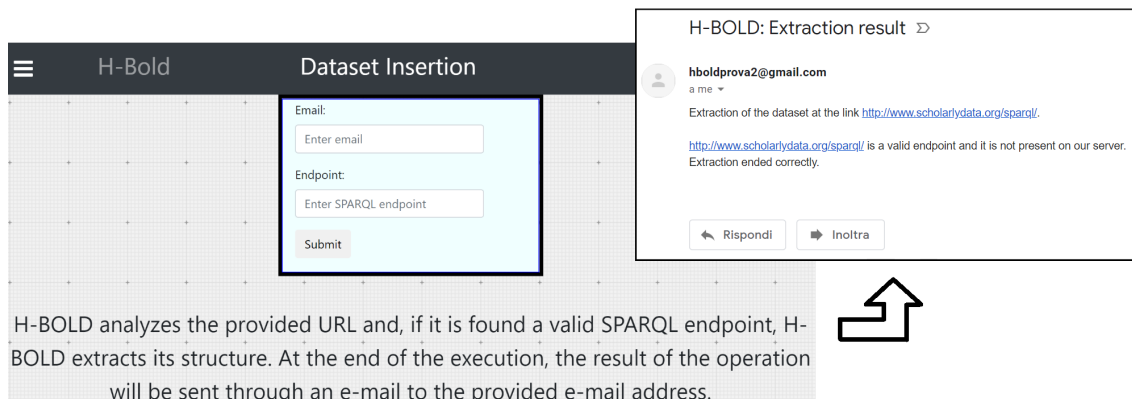
Figure 3: Interface for the insertion of a SPARQL endpoint and the e-mail of the successfully extraction

layer and the presentation layer. The server layer was set up to query three new open data portals. In particular, now, H-BOLD search for new endpoints in the following portals:

- European Data Portal[8]. It contains open data published over the different european countries, regions and local administrations open data portals;
- EU Open Data Portal[9]. It holds the data produced by the different organization of the European Union;
- IO Data Science of Paris[10]. Born with the idea of creating new synergies between data, it contains a metadata description of several LD.

The portals were deeply explored in order to gain knowledge over their content. Initially, we tried to produce three ad-hoc queries for extracting the maximum amount possible of SPARQL endpoints but then we found that the query presented in Listing 1 perfectly fits all the portals.

Listing 1: Query sent to the open data portals to extract a list of SPARQL endpoints

```
PREFIX  dcat:  <http://www.w3.org/ns/dcat#>
PREFIX  dc:  <http://purl.org/dc/terms/>

SELECT  ?dataset ?title ?url
WHERE  {
    ?dataset  a  dcat:Dataset .
    ?dataset  dc:title  ?title .
    ?dataset  dcat:distribution  ?distribution .
    ?distribution  dcat:accessURL  ?url .
    filter(regex(?url, 'sparql')).
}
```

With this research, we discovered 65 SPARQL endpoints on the European Data Portal, 9 SPARQL endpoints on the EU Open Data Portal and 15 SPARQL endpoints on the IO Data Science of Paris. Some endpoints were already present in H-BOLD, therefore we were effectively able to increment the number of endpoints in our collection by 70 units. As a result, the number of endpoints listed in H-BOLD raised from 610 to 680. Since some of them are not working or are not compatible with the index extraction phase of H-BOLD, we were able to index and expose the structure of 20 new datasets raising the number of indexed endpoints from 110 to 130.

---

[8]https://www.europeandataportal.eu/
[9]https://data.europa.eu/euodp/en/home/
[10]https://io.datascience-paris-saclay.fr

3.4 Manual insertion of new endpoints

Even with the crawling of SPARQL endpoint from open data portals, we are not able to reach, index and expose every SPARQL endpoint available on internet. Therefore, to further increase the number of indexed datasets, we integrated in H-BOLD a procedure through which the user is able to upload the URL of a SPARQL endpoint and to see, this new dataset listed among the others in the H-BOLD dataset list (we tested this procedure over some open datasets [6]). Since the index extraction procedure can be time-consuming, the user is asked to provide an e-mail address so that the system can notify he/she about the status of the extraction. At the end of the extraction, the e-mail address is deleted, since we do not want to keep person data, while the dataset is added to the list of available datasets. In this way, we enlarge the list of LD that H-BOLD is able to visualize and we made H-BOLD more responsive to the user requests. Figure 3 contains the interface for uploading the URL of a SPARQL endpoint and the email that will be sent to the users in case the index extraction procedure is successful.

3.5 New visualizations

The previous version of H-BOLD adopted only graphs for displaying the information (as shown in Figure 2) and in some occasion they were not suited for extracting the maximum of the information. For instance, the number of instances belonging to a class is hard to extrapolate from a graph and it is also hard to understand which classes have been aggregate in what cluster. Moreover, when the number of classes and relations is high, it is hard to understand the various connection. For this reason we implemented four supplementary visualization layouts, three for the Cluster Schema and one for the Schema Summary. We drew inspiration from an extensive analysis over other LD visualization tools [14]. The new visualizations for the Cluster Schema allow displaying together the clusters and the classes, providing users with a complete high-level overview of the dataset. The new visualization for the Schema Summary allow to better understanding the inter-connections among classes and the incoming and out-going properties.

3.5.1 Treemap visualizations of the Cluster Schema. Treemaps are an alternative way of visualising the hierarchical structure of a Cluster Schema while also displaying quantities for each cluster and each class within the cluster via area size. Each cluster is assigned to a rectangle area with a specific color and their classes rectangles nested inside of it. When a quantity is assigned to

Figure 4: Treemap visualization of the Cluster Schema

a class, its rectangle area size is displayed in proportion to that quantity and to the other quantities within the same cluster in a part-to-whole relationship. Also, the area size of the cluster is the total of its classes. If no quantity is assigned to a class, then its area is divided equally amongst the other classes within its cluster. The treemap built over the instance's count, shown in Figure 4), highlights the classes with the higher number of instances, the size of the clusters and the predominance of some classes in terms of instances.

Figure 5: Sunburst visualization of the Cluster Schema

Figure 6: Circle Pack visualization of the Cluster Schema

represented as a circle and its sub-branches are represented as circles inside it. Similarly to the rectangles in the Treemap, the circles might have different dimensions. As displayed in Figure 6, the inner circles represent the classes, while the intermediate circles represent the clusters, an external circle represents the entire dataset. In some cases, a cluster can contain only one class.

Figure 7: Hierarchical Edge graph visualization of the Schema Summary

3.5.2 Sunburst visualizations of the Cluster Schema. The Sunburst Chart visualization (Figure 5) shows the hierarchy through a series of rings, that is sliced for each category node. The inner ring represents the clusters while the outer ring shows the classes grouped by the clusters.

3.5.3 Circle Pack visualizations of the Cluster Schema. The Circle Packing (Figure 6) is a variation of a Treemap that uses circles instead of rectangles. Containment within each circle represents a level in the hierarchy: each branch of the tree is

3.5.4 Hierarchical edge bundling visualization for the Schema Summary. Hierarchical edge bundling is a method developed by

Holten in 2006 [11] for allowing to visualize adjacency relations between entities organized in a hierarchy. The idea is to bundle the adjacency edges together to decrease the clutter usually observed in complex networks. This data visualisation method allows to check connections between leaves (classes in our case) of a hierarchical network.

Using the Hierarchical Edge Bundling layout, the classes are displayed over an invisible circumference and the properties are arcs within the circumference. This layout is perfectly suited for understanding links within the classes and the domain and range of the properties that connect the classes. As an example, in Figure 7, the node in bold (*Event*) is the class of interest, the node in green (*Situation*) is the *rdfs:Range* class of the properties that connects it to the class of interest and the nodes in red (*Vevent, SessionEvent, ConferenceSeries and InformationObject*) are the *rdfs:Domain* classes of the properties that connect them to the class of interest.

4 RELATED WORK

In [7], a model for building, visualizing, and interacting with hierarchically organized numeric and temporal LD has been proposed. This method has been implemented in a framework for hierarchical charting and exploration of LD called "rdf:SynopsViz" [8]. The tool is available online[11] for the exploration of a single dataset and the hierarchical charting available are mainly focused on numeric or datetime properties. The exploration of new SPARQL endpoint is not working[12].

In [18], a hierarchical co-clustering approach over LD has been proposed. It simultaneously groups links and entity classes exploiting measures of intra-link and intra-class similarity. This approach has been implemented in a LD browser called CoClus. Although extensive evaluations have been carried out and demonstrated that the approach provides useful support for entity exploration, the browser CoClus is not directly accessible online, therefore it is impossible to compare it w.r.t. H-BOLD. Another interesting and recent tool for interactive LD visualization is S-Paths [9]. It can display multiple views on RDF resource sets and supports browsing over the Web of Data. It is able to show different properties along paths in the graph. Users can navigate between different resource sets by selecting a subset or pivoting.

5 CONCLUSION AND FUTURE WORK

In this paper, we presented H-BOLD, a tool for multilevel interactive visual exploration of Big LD that has been enhanced with new visualizations: TreeMap, Sunburst Chart and Circle Packing. H-BOLD has been tested on 130 Big LD showing good performances.

In the next future, we intend to raise the number of endpoints indexed in H-BOLD by improving the index extraction procedure and by querying new repositories that collect SPARQL endpoints metadata. Moreover, we intend to evaluate the effectiveness of H-BOLD as a visualization tool through a survey involving different kinds of LD consumers: practitioners, unskilled users, domain experts.

6 ACKNOWLEDGMENTS

REFERENCES

[1] Fabio Benedetti, Sonia Bergamaschi, and Laura Po. 2014. Online Index Extraction from Linked Open Data Sources. In *LD4IE@ISWC (CEUR Workshop Proceedings)*, Vol. 1267. CEUR-WS.org, 9–20.

[2] Fabio Benedetti, Sonia Bergamaschi, and Laura Po. 2015. Exposing the Underlying Schema of LOD Sources. In *IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, WI-IAT 2015, Singapore, December 6-9, 2015 - Volume I*. IEEE Computer Society, 301–304. https://doi.org/10.1109/WI-IAT.2015.99

[3] Fabio Benedetti, Sonia Bergamaschi, and Laura Po. 2015. LODeX: A Tool for Visual Querying Linked Open Data. In *International Semantic Web Conference (Posters & Demos) (CEUR Workshop Proceedings)*, Vol. 1486. CEUR-WS.org.

[4] Fabio Benedetti, Sonia Bergamaschi, and Laura Po. 2015. Visual Querying LOD sources with LODeX. In *K-CAP*. ACM, 12:1–12:8.

[5] Fabio Benedetti, Laura Po, and Sonia Bergamaschi. 2014. A Visual Summary for Linked Open Data sources. In *Proceedings of the ISWC 2014 Posters & Demonstrations Track a track within the 13th International Semantic Web Conference, ISWC 2014, Riva del Garda, Italy, October 21, 2014 (CEUR Workshop Proceedings)*, Matthew Horridge, Marco Rospocher, and Jacco van Ossenbruggen (Eds.), Vol. 1272. CEUR-WS.org, 173–176. http://ceur-ws.org/Vol-1272/paper_136.pdf

[6] Domenico Beneventano, Sonia Bergamaschi, Luca Gagliardelli, and Laura Po. 2015. Open Data for Improving Youth Policies. In *KEOD 2015 - Proceedings of the International Conference on Knowledge Engineering and Ontology Development, part of the 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2015), Volume 2, Lisbon, Portugal, November 12-14, 2015*, Ana L. N. Fred, Jan L. G. Dietz, David Aveiro, Kecheng Liu, and Joaquim Filipe (Eds.). SciTePress, 118–129. https://doi.org/10.5220/0005625401180129

[7] Nikos Bikakis, George Papastefanatos, Melina Skourla, and Timos Sellis. 2017. A hierarchical aggregation framework for efficient multilevel visual exploration and analysis. *Semantic Web* 8, 1 (2017), 139–179. https://doi.org/10.3233/SW-160226

[8] Nikos Bikakis, Melina Skourla, and George Papastefanatos. 2014. rdf: SynopsViz - A Framework for Hierarchical Linked Data Visual Exploration and Analysis. In *The Semantic Web: ESWC 2014 Satellite Events - ESWC 2014 Satellite Events, Anissaras, Crete, Greece, May 25-29, 2014, Revised Selected Papers (Lecture Notes in Computer Science)*, Valentina Presutti, Eva Blomqvist, Raphaël Troncy, Harald Sack, Ioannis Papadakis, and Anna Tordai (Eds.), Vol. 8798. Springer, 292–297. https://doi.org/10.1007/978-3-319-11955-7_37

[9] Marie Destandau. 2019. Interactive visualisation techniques for the web of data. *The Web Conference 2019 - Companion of the World Wide Web Conference, WWW 2019* (2019), 17–21. https://doi.org/10.1145/3308560.3314189

[10] Sébastien Ferré. 2017. Sparklis: An expressive query builder for SPARQL endpoints with guidance in natural language. *Semantic Web* 8, 3 (2017), 405–418. https://doi.org/10.3233/SW-150208

[11] Danny Holten. 2006. Hierarchical Edge Bundles: Visualization of Adjacency Relations in Hierarchical Data. *IEEE Transactions on Visualization and Computer Graphics* 12 (2006). Issue 5.

[12] Steffen Lohmann, Vincent Link, Eduard Marbach, and Stefan Negru. 2016. Extraction and Visualization of TBox Information from SPARQL Endpoints. In *Proceedings of the 20th International Conference on Knowledge Engineering and Knowledge Management (EKAW 2016) (LNAI)*, Vol. 10024. Springer, 713–728.

[13] Laura Po. 2018. High-level Visualization Over Big Linked Data. In *Proceedings of the ISWC 2018 Posters & Demonstrations, Industry and Blue Sky Ideas Tracks co-located with 17th International Semantic Web Conference (ISWC 2018), Monterey, USA, October 8th - to - 12th, 2018 (CEUR Workshop Proceedings)*, Marieke van Erp, Medha Atre, Vanessa López, Kavitha Srinivas, and Carolina Fortuna (Eds.), Vol. 2180. CEUR-WS.org. http://ceur-ws.org/Vol-2180/paper-50.pdf

[14] Laura Po, Nikos Bikakis, Federico Desimoni, and George Papastefanatos. 2020. *Linked Data Visualization*. Morgan & Claypool Publishers. to appear.

[15] Laura Po and Davide Malvezzi. 2018. Community Detection Applied on Big Linked Data. *J. UCS* 24, 11 (2018), 1627–1650. http://www.jucs.org/jucs_24_11/community_detection_applied_on

[16] Georgia Troullinou, Haridimos Kondylakis, Evangelia Daskalaki, and Dimitris Plexousakis. 2015. RDF Digest: Efficient Summarization of RDF/S KBs. In *ESWC (Lecture Notes in Computer Science)*, Vol. 9088. Springer, 119–134.

[17] Fabio Viola, Luca Roffia, Francesco Antoniazzi, Alfredo D'Elia, Cristiano Aguzzi, and Tullio Salmon Cinotti. 2018. Interactive 3D Exploration of RDF Graphs through Semantic Planes. *Future Internet* 10, 8 (2018). https://doi.org/10.3390/fi10080081

[18] Liang Zheng, Yuzhong Qu, Xinqi Qian, and Gong Cheng. 2018. A hierarchical co-clustering approach for entity exploration over Linked Data. *Knowl.-Based Syst.* 141 (2018), 200–210. https://doi.org/10.1016/j.knosys.2017.11.017

---

[11]http://synopsviz.imis.athena-innovation.gr/
[12]Tests have been performed on Jan 3rd, 2020