Building MD Analytical Stars Using a Visual Linked Data Query Builder

Victoria Nebot(✉) and Rafael Berlanga

Computer Languages and Systems, Universitat Jaume I,
Campus Riu Sec s/n, 12071 Castellón, Spain
{romerom,berlanga}@uji.es

Abstract. In this paper we present the foundations and a prototype tool for Linked Data (LD) analysis and visualization. The goal is to provide users with a tool enabling them to easily analyze Linked Data without knowledge of the SPARQL query language. The success of the multidimensional (MD) model for data analysis has been mainly due to its simplicity. The MD model views data in terms of dimensions and measures. Therefore, we propose the notion of MD analytical stars as the foundation to query LD. These stars are MD conceptual patterns that summarize LD (i.e., dimensions and measures). We have developed a tool that enables the user to graphically build MD analytical stars as queries by suggesting possible dimensions and measures from a LD set. The query is automatically translated to SPARQL and executed. Visualization of results as charts, tables, etc. is performed using web-based technologies. We show the prototype tool and a running example.

Keywords: Linked Data · RDF · Multidimensional models · Linked data querying

1 Introduction

During the last years, communities from different areas have published data in the cloud of LD following the publication guidelines, providing the basis for creating and populating the Web of Data. Currently, there are approximately 74 billion triples over 1000 datasets[1].

The increasing availability of these semi-structured and semantically enriched datasets has prompted the need for new tools able to explore, query, analyze and visualize these semi-structured data [5]. While several different tools such as graph-based query builders, semantic browsers and exploration tools [2–4,8] have emerged to aid the user in querying, browsing and exploring LD, these approaches have a limited ability to summarize, aggregate and display data in the form that a scientific or business user expects, such as tables and graphs. Moreover, they fall short when it comes to provide the user an overview of the data that may be of interest from an analytical viewpoint.

[1] http://stats.lod2.eu.

A. Fred et al. (Eds.): IC3K 2014, CCIS 553, pp. 198–211, 2015.
DOI: 10.1007/978-3-319-25840-9_13

LD constitutes a valuable source of knowledge worth exploiting using analytical tools. Business Intelligence (BI) uses the MD model to view and analyze data in terms of dimensions and measures, which seems the most natural way to arrange data. BI has traditionally been applied to internal, corporate and structured data, which is extracted, transformed and loaded (ETL) into a pre-defined and static MD model. The relational implementation of the MD data model is typically a star schema. The dynamic and semi-structured nature of LD poses several challenges to both potential analysts and current BI tools. On one hand, exploring the datasets using the available browsers and tools to find MD patterns is cumbersome due to the semi-structured nature of the data and the lack of support for obtaining summaries of the data. Moreover, as the datasets are dynamic their structure may change or evolve, making the one-time MD design approach unfeasible.

In this paper, we propose MD analytical stars as the foundation to query LD. A MD analytical star is a MD star-shaped pattern at the concept level that encapsulates an interesting MD analysis [14]. The star is focused on a subject of analysis and is composed by one or several measures (i.e., measurable attributes on which calculations can be made) and dimensions (i.e., the different analytical perspectives). These stars reflect relevant patterns in the dataset, as the measures and dimensions that compose them are calculated following a statistical approach [15]. To ease the composition of MD analytical stars we have developed a web-based prototype tool that allows the user to easily compose them by selecting suggested dimensions and measures for a given subject of analysis. With this user-friendly query builder we are freeing the user from the cumbersome task of browsing, exploring and building queries in specific languages to find interesting analytical patterns in large LD sets. Moreover, the tool is able to translate the user graphical query to SPARQL, execute it and display the results using different charts and diagrams.

We summarize our contribution as follows:

- We define the concept of MD analytical star as a mapping of the MD model to LD. That is, we identify the subject of analysis, dimensions and measures that compose a MD analytical star in LD.
- We introduce the notion of aggregation power for the dimensions and measures and make an estimation to filter dimensions and measures according to this score.
- We have developed a user-friendly web-based tool that allows users to query and visualize results from LD sets by dynamically building MD analytical stars.

The structure of the paper is as follows. In Sect. 2 we review the literature related to the problem of analyzing LD. Section 3 presents the main foundations that underlie our approach. In Sect. 4 we present a model for MD analytical stars over LD sources. Section 5 summarizes how dimensions and measures are calculated from LD. Section 6 explains the prototype implemented and shows a running example. Finally, Sect. 7 gives some conclusions and future work.

2 Related Work

We have performed a thorough review on the related literature to find out that the majority of approaches use querying, exploration and only light-weight analytics over LD.

For querying LD, SPARQL has become the de-facto standard. However, directly querying a dataset using SPARQL interface cannot be considered an end-user task as it requires familiarity with its syntax and the structure of the underlying data. Graph-based query builders such as [3] can help users build triple patterns by using auto-completion to express queries. However, users do not always have explicit queries upfront, but need to explore the available data first in order to find out what information might be interesting to them. Sgvizler[2] allows to render results of SPARQL queries as charts, maps, etc. However, it requires SPARQL knowledge and focuses only on the visualization part.

The review in [5] about visualization and exploration of LD concludes that most of the tools are designed for technical users and do not provide an overview or summary of the data.

LD browsers such as [2,4,19] are designed to display one entity at a time and do not support the user in aggregation tasks. Most of them use faceted filtering to better guide the user in exploration tasks. However, the user gets overview of only a small part of the dataset. On the other hand, browsers such as [17,18] provide a more powerful browsing environment, but are tailored to a specific application.

Graph-based tools such as RDF-Gravity[3], IsaViz[4] or Relfinder [8] provide node-link visualizations of the datasets and the relationships between them. Although this approach can obtain a better understanding of the data structure, graph visualization does not scale well to large datasets.

The CODE Query Wizard and Vis Wizard developed under the CODE project[5] are a web-based visual analytics platform that enables non-expert users to easily perform exploration and lightweight analytic tasks on LD. Still, the user has to browse the data to find interesting analytical queries. Payola [12] is a framework that allows any expert user to access a SPARQL endpoint, perform analysis using SPARQL queries and visualize the results using a library of visualizers.

We claim that existing tools for exploration and analysis of LD provide little or no support for summaries so that the user can have an idea of the structure of the dataset and the parts that seem more interesting for analysis. In that line, we have also looked into approaches that provide graph summaries over LD using different techniques such as bisimulation and clustering [1,10]. However, these graph summaries are produced without an analytical focus, therefore, the resulting summaries may not be useful for analysis purposes.

[2] http://dev.data2000.no/sgvizler/.

[3] http://semweb.salzburgresearch.at/apps/rdf-gravity/.

[4] http://www.w3.org/2001/11/IsaViz/.

[5] http://code-research.eu/.

Recently, there have been some attempts to analyze LD that go beyond querying and browsing. [14] proposes MD analysis over LD under the OWL formalism. However, most LD sets lack this semantic layer. Other approaches [6,9] have proposed MD analysis over LD relying on the previous manual annotation of the MD elements (dimensions and measures) and using previously defined MD vocabularies.

3 Background

In this section, we review background concepts on which our approach is based on.

3.1 Linked Data

LD is a set of common practices and general rules to contribute to the Web of Data [7]. The basic principles are that each entity should be assigned a unique URL identifier, the identifiers should be dereferenceable by HTTP and the entity representations should be interlinked together to form a global LD cloud.

The most adopted standard to implement the Web of Data is RDF [13], which allows us to make statements about entities. It assumes data modeled as triples with three components: subject, predicate and object. We consider only valid RDF triples using URIs (U), blank nodes (B) and literals (L). These triples can also be viewed as graphs, where vertices correspond to subjects and objects, while labeled edges represent the triples themselves. SPARQL [16] has become the standard for querying RDF data and it is based on the specification of triple patterns.

In RDF there is no technical distinction between the schema and the instance data, even though it provides terminology to express class membership or categorization (`rdf:type`). The RDFS extension allows to create taxonomies of classes and properties. It also extends definitions for some of the elements of RDF, for example it sets the domain and range of properties and relates the RDF classes and properties into taxonomies using the RDFS vocabulary. OWL extends RDFS and allows for expressing further schema definitions in RDF. The formal semantics of RDFS and OWL enrich RDF with implicit information that can be reasoned over. Throughout the paper, we refer both to the explicit and implicit triples, which have been derived using some reasoning mechanism. We use the naming convention of OWL referring to classes, properties and individuals to homogenize terminology.

3.2 Multidimensional Models

The MD model is the conceptual abstraction mostly used in BI. The observations or facts are analyzed in terms of dimensions and measures [11]. They focus on a subject of analysis (e.g., sales) and define a series of dimensions or different analysis perspectives (e.g., location, time, product), which provide contextual information. Facts are aggregated in terms of a series of measures (e.g., average sales). As a result, analysts are able to explore and query the resulting data cube applying OLAP operations. A typical query would be to display the evolution of the sales during the current year of personal care products by city.

BI has traditionally been applied to internal, corporate and structured data, which is extracted, transformed and loaded into a pre-defined and static MD model. The relational implementation of the MD data model is typically a star schema, where the fact table containing the summarized data is in the center and is connected to the different dimension tables by means of functional relations.

4 MD Analytical Stars

In this section we explain how we model MD analytical stars from LD.

We formalize the representation of an RDF graph using graph notation.

Definition 1. *(RDF graph) An RDF graph G is a labeled directed graph $G = \langle V, E, \lambda \rangle$ where:*

- *V is the set of nodes, let V^0 denote the nodes in V having no outgoing edge, and let $V^{>0} = V \backslash V^0$;*
- *$E \subseteq V \times V$ is the set of directed edges;*
- *$\lambda : V \cup E \rightarrow U \cup B \cup L$ is a labeling function such that $\lambda_{|V}$ is injective, with $\lambda_{|V^0} : V^0 \rightarrow U \cup B \cup L$ and $\lambda_{|V^{>0}} : V^{>0} \rightarrow U \cup B$, and $\lambda_{|E} : E \rightarrow U$.*

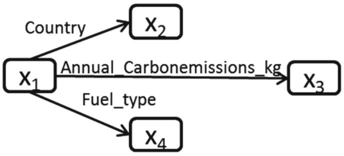
Typical analysis usually involves investigating a set of particular facts according to relevant criteria (dimensions) and measurable attributes (measures). Here, we use the notion of basic graph pattern (BGP) queries, which is a well-known a subset of SPARQL. A BGP is a set of triple patterns, where each triple has a subject, predicate and object, some of which can be variables. We are specially interested in rooted BGP queries, as they resemble the star-shaped pattern typical of MD analysis.

Definition 2. *(Rooted query) Let q be a BGP query, $G = \langle V, E, \lambda \rangle$ its graph and $v \in V$ a node that is a variable in q. The query q is rooted in v iff G is a connected graph and any other node $v' \in V$ is reachable from v following the directed edges in E.*

Example 1. (Rooted query) The query q is a rooted BGP query, with x_1 as root node.

$$\begin{aligned} q(x_1, x_2, x_3, x_5) \text{ :- } & x_1 \text{ Annual\_Carbonemissions\_kg } x_3, \\ & x_1 \text{ Country } x_2, \\ & x_1 \text{ Fuel\_type } x_4 \end{aligned}$$

The query's graph representation below shows that every node is reachable from the root x_1.



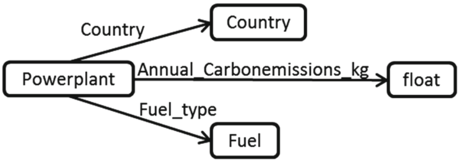
Even though rooted queries express data patterns by means of the predicate chains, these are still vague as the variable nodes can match any element in $U \cup B \cup L$. To narrow down the scope of the patterns we define the notion of typified rooted queries as follows:

Definition 3. *(Typified rooted query) A typified rooted query q' is a rooted query with graph $G = \langle V, E, \lambda \rangle$ where each variable node $v_x \in V$ has an associated class or datatype. That is, each variable v_x has an outgoing edge (v_x, v_y) such that $\lambda((v_x, v_y)) =$ **`rdf:type`** and $\lambda(v_y) \in U$ and v_y has an outgoing edge (v_y, v_z) such that $\lambda((v_y, v_z)) =$ **`rdf:type`** and $\lambda(v_z) \in \{$**`rdfs:Class`**, **`rdfs:Datatype`**$\}$.*

Example 2. (Typified rooted query) The previous query q can be typified as follows:

$$
\begin{aligned}
q(x_1,&x_2,x_3,x_5) \text{ :- } x_1 \text{ rdf:type Powerplant, Powerplant rdf:type rdfs:Class,}\\
&x_1 \text{ Country } x_2,\ x_2 \text{ rdf:type Country, Country rdf:type rdfs:Class,}\\
&x_1 \text{ Annual\_Carbonemissions\_kg } x_3,\\
&x_3 \text{ rdf:type xsd:float, xsd:float rdf:type rdfs:Datatype,}\\
&x_1 \text{ Fuel\_type } x_4,\ x_4 \text{ rdf:type Fuel, Fuel rdf:type rdfs:Class}
\end{aligned}
$$

From now on, we omit the type edges and represent typified rooted queries with the (data)type's name in the variable node.



It is immediate to see that a typified rooted query is composed by a set of typified paths that go from the root to a sink node (node with no outgoing edges). The root node represents a class and the sink node represents either a class or a datatype. We formalize this notion next:

Definition 4. *(Typified path) Given a typified rooted query q with x_1 as root node and graph $G = \langle V, E, \lambda \rangle$, a typified path is a sequence $p = c_1 - r_1 - c_2 - r_2 - ... - r_{n-1} - c_f$ where $\lambda(x_1) = c_1$ is the root class, c_f is a sink class or datatype, every r_i is a property and every c_i has an associated class.*

Example 3. (Typified path) In the previous query q we can identify the following typified paths:

(Powerplant, Country, Country)
(Powerplant, Annual_Carbonemissions_kg, float)
(Powerplant, Fuel_type, Fuel)

In MD modeling it is important the many-to-one relation between facts and dimensions to ensure aggregation power. That is, one fact must be associated with one dimension value, whereas a dimension value can and should be associated to multiple facts. In our LD scenario, we define the aggregation power of a typified path as follows:

Definition 5. *(Aggregation power) Given a typified path $p = c_1 - r_1 - c_2 - r_2 - ... - r_{n-1} - c_f$, the aggregation power is calculated as the ratio between the number of different individuals (or literals) of the sink class (or datatype) c_f that satisfy the path and the number of individuals of the root class c_1 that satisfy the path.*

Example 4. (Aggregation power) Given the following paths, we calculate the aggregation power as follows:

$$\text{(Powerplant}_{(74561)}\text{, Country, Country}_{(200)}\text{)} \rightarrow \frac{200}{74561} = 0.0027$$
$$\text{(Powerplant}_{(19796)}\text{, Fuel\_type, Fuel}_{(30)} \rightarrow \frac{30}{19796} = 0.0015$$

Notice that an aggregation power closer to 0 exhibits a high aggregation capacity, meaning that the sink individuals act as categories for the root individuals of the path. Aggregation power closer to 1 means low aggregation power of the path.

We are now ready to introduce MD analytical stars. For this, we make use of traditional data warehousing terminology. We use the notion of *classifier* to denote the level of data aggregation, that is, the classifier defines the dimensions according to which the facts will be analyzed. The *measure* allows obtaining values to be aggregated using *aggregation functions.*

Definition 6. *(MD analytical star) Given an RDF graph $G = \langle V, E, \lambda \rangle$, a MD analytical star rooted in the node $x \in V$ is a triple: $S = \langle c(x, d_1, ..., d_n), m(x, v), \bigoplus \rangle$ where:*

- *$c(x, d_1, ..., d_n)$ is a typified query rooted in the node r_c of its graph G_c, with $\lambda(r_c) = x$ and each path $x - ... - d_i$ is a typified path. This is the classifier of x w.r.t. the n dimensions $d_1, ..., d_n$. The node x is the subject of analysis.*
- *$m(x, v)$ is a typified query rooted in the node r_m of its graph G_m, with $\lambda(r_m) = x$. This query is only composed by a typified path $x - ... - v$[6]. This is called the measure of x.*
- *$\bigoplus$ is an aggregation function over a set of values, that is, the aggregator for the measure of x w.r.t. its classifier.*

[6] For the sake of simplicity, we assume that an MD analytical star has only one measure.

– *Each of the typified paths of the classifier has an aggregation power below a threshold δ.*

Notice that typified rooted queries (and therefore, typified paths) are the building block to suggest MD analytical stars.

Example 5. (MD analytical star) The MD analytical star below asks for the average of annual carbon emission of powerplants, classified by country and fuel type.

$$\langle c(x, x_1, x_3), m(x, x_4), average \rangle$$

where the classifier and measure queries are:

$$\begin{aligned} c(x, x_1, x_3) \; &:- \; x \text{ rdf:type Powerplant,} \\ &\quad\; x \text{ Country } x_1,\; x_1 \text{ rdf:type Country,} \\ &\quad\; x \text{ Fuel\_type } x_2,\; x_2 \text{ rdf:type Fuel} \\ m(x, x_4) \; &:- \; x \text{ rdf:type Powerplant,} \\ &\quad\; x \text{ Annual\_Carbonemissions\_kg } x_4,\; x_4 \text{ rdf:type xsd:float} \end{aligned}$$

The answer to an MD analytical star is a set of tuples of dimension values found in the answer of the classifier query, together with the aggregated result of the measure query. Therefore, it can be represented as a cube of n dimensions, where each cell contains the aggregated measure.

5 Calculating Dimensions and Measures

MD analytical stars are composed by the classifier and measure typified queries rooted in a potential class acting as subject of analysis. These typified queries are composed by typified paths with a certain aggregation power. For space restrictions, we omit the process of calculating the set of typified paths from a LD. The method is based on probabilistic graphical models and we make use of the statistics about instance data to generate the paths. We refer the reader to Sect. 5 of [15] for details and to Sect. 6 for checking the experimental evaluation. As a result, a set of typified paths with aggregation power below a threshold δ are obtained.

Paths are classified into dimensions and measures based on the aggregation power of the path and the type of the sink node. Paths ending in numeric datatypes (i.e., `xsd:integer`, `xsd:float`, `xsd:double`, etc.) and with low aggregation power are considered measures, whereas the rest of paths (i.e., paths ending in classes or datatypes with high aggregation power) are considered dimensions.

The set of calculated dimensions and measures from a LD dataset is used by the prototype tool to help the user build MD analytical stars. In the next section, we show the implemented prototype along with a running example.

6 Prototype Tool

The current prototype for building and executing MD analytical stars over LD sets has been implemented as a client/server web application. The architecture is shown in Fig. 1. We distinguish the client, the server, an external LD endpoint, and the catalogue of calculated dimensions and measures. For the client we use AJAX and HTML5/CSS3. The server is implemented as a RESTFUL API in PHP. It has three main tasks: (1) handle requests from the user about the dimensions and measures of the catalogue, (2) act as a proxy to send queries to the LD endpoint and (3) process the results of the endpoint. The external LD endpoint is a SPARQL endpoint that we access to execute the MD queries. The catalogue is the application data, that is, the dimensions and measures calculated for the LD set. We have implemented the catalogue as an independent RESTful web service to make it portable.

This prototype helps the user build MD analytical stars by suggesting possible dimensions and measures for a specific subject of analysis (steps 1–4). Moreover, it is completely functional, as the queries graphically built by the user are automatically translated to SPARQL queries over the dataset endpoint, which returns the results that can be either displayed or exported in different formats (steps 5–7).

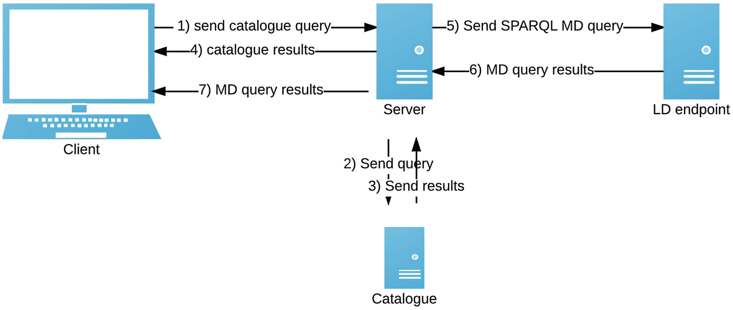


Fig. 1. Prototype architecture.

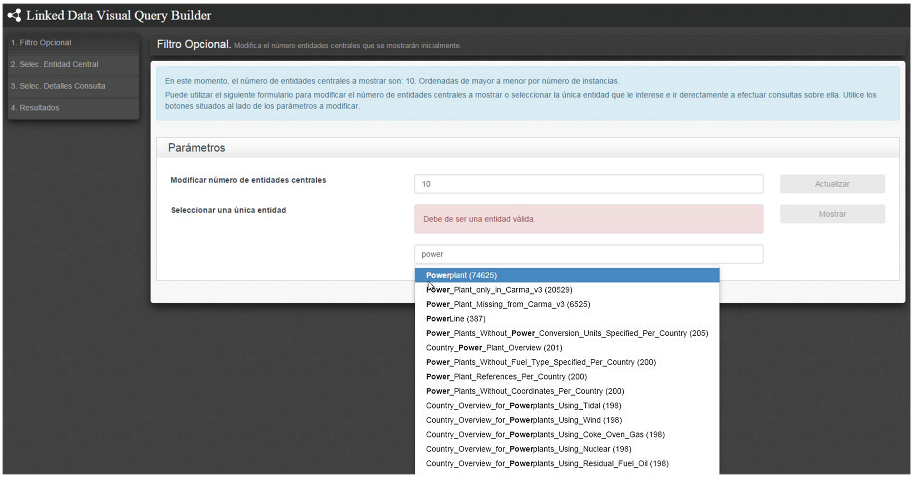
For demonstration purposes, we have selected the Enipedia[7] LD dataset, which is an initiative aimed at providing a collaborative environment through the use of wikis and the Semantic Web for energy industry issues. The dataset provides energy-related data from different open data sources structured and linked in RDF. The dataset contains around 5M triples.

Next, we show the different functionality of the prototype. The home page is shown in Fig. 2. The left navigation menu shows the different steps to perform a MD query, which are: (1) show most interesting subjects of analysis, (2) select

[7] http://enipedia.tudelft.nl/wiki/Main_Page.

one, (3) select measures and dimensions and (4) show results. We will go through all of them by means of a running example.

In the home page the user can ask to show the n most interesting subjects of analysis if (s)he has no knowledge of the dataset, or directly type the name of a class, which comes with the autocompletion feature. Interestingness is measured by the number of individuals of the class. In Fig. 2 we select *powerplant* as subject of analysis as we are interested in analyzing the annual carbon emission rates of powerplants from different perspectives.


Linked Data Visual Query Builder
1. Filtro Opcional
2. Selec. Entidad Central
3. Selec. Detalles Consulta
4. Resultados
Actualizar
Mostrar

Filtro Opcional. Modifica el número entidades centrales que se mostrarán inicialmente.

En este momento, el número de entidades centrales a mostrar son: 10. Ordenadas de mayor a menor por número de instancias.
Puede utilizar el siguiente formulario para modificar el número de entidades centrales a mostrar o seleccionar la única entidad que le interese e ir derectamente a efectuar consultas sobre ella. Utilice los botones situados al lado de los parámetros a modificar.

Parámetros

Modificar número de entidades centrales: 10

Seleccionar una única entidad: Debe de ser una entidad válida.

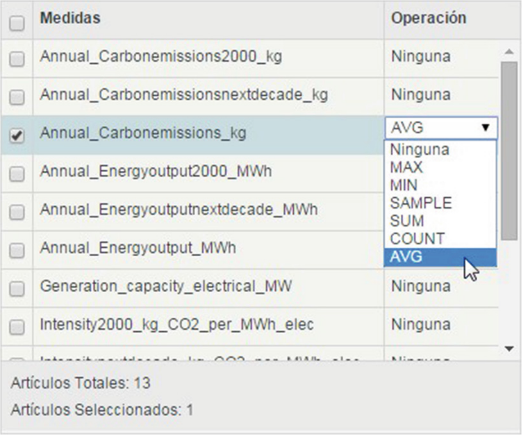
power

- **Power**plant (74625)
- **Power**_Plant_only_in_Carma_v3 (20529)
- **Power**_Plant_Missing_from_Carma_v3 (6525)
- **Power**Line (387)
- **Power**_Plants_Without_**Power**_Conversion_Units_Specified_Per_Country (205)
- Country_**Power**_Plant_Overview (201)
- **Power**_Plants_Without_Fuel_Type_Specified_Per_Country (200)
- **Power**_Plant_References_Per_Country (200)
- **Power**_Plants_Without_Coordinates_Per_Country (200)
- Country_Overview_for_**Power**plants_Using_Tidal (198)
- Country_Overview_for_**Power**plants_Using_Wind (198)
- Country_Overview_for_**Power**plants_Using_Coke_Oven_Gas (198)
- Country_Overview_for_**Power**plants_Using_Nuclear (198)
- Country_Overview_for_**Power**plants_Using_Residual_Fuel_Oil (198)

Fig. 2. Home page. The user selects to display the n most interesting subjects of analysis or types one.

The next step consists in selecting measures and dimensions for the subject of analysis. These are suggested by the prototype and the user is in charge only of selecting the ones required. Figure 3 shows the panel where all available measures for the subject *powerplant* are displayed and the user can select one or more by ticking the box and selecting the appropriate aggregation function. In this case we select *Annual_Carbonemissions2000_kg* and *average* as the aggregation function.

The process of selecting the dimensions is shown in Figs. 4 and 5. The left panel of Fig. 4 shows the possible dimensions. In this case, the user selects two dimensions, *Country* and *Fuel*, meaning that the analysis (s)he has in mind consists in displaying the average carbon emission levels of the powerplants by country and the type of fuel of each powerplant. The right panel serves the purpose of disambiguating a selected dimension. Notice that a dimension is defined by a typified path from the subject of analysis to a sink class. By only selecting the sink class as dimension there can be ambiguities as a sink class can be reached by different paths and the properties of the paths may give different

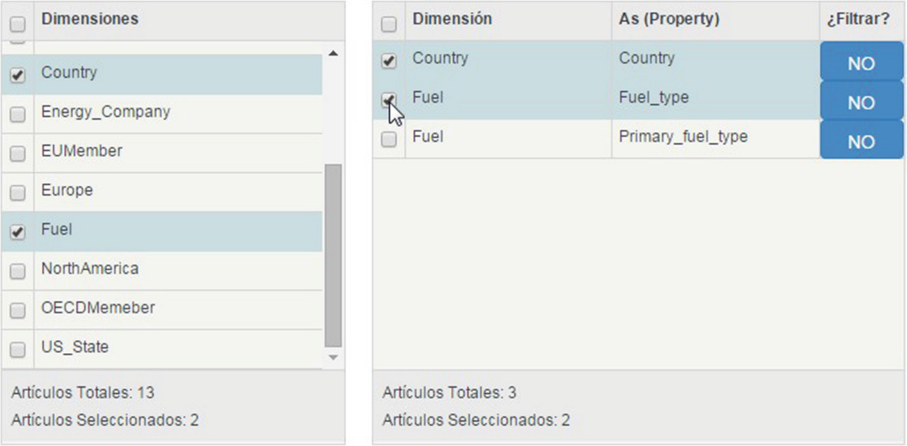


□	Medidas	Operación
□	Annual_Carbonemissions2000_kg	Ninguna
□	Annual_Carbonemissionsnextdecade_kg	Ninguna
☑	Annual_Carbonemissions_kg	AVG
□	Annual_Energyoutput2000_MWh	
□	Annual_Energyoutputnextdecade_MWh	
□	Annual_Energyoutput_MWh	
□	Generation_capacity_electrical_MW	Ninguna
□	Intensity2000_kg_CO2_per_MWh_elec	Ninguna

Artículos Totales: 13
Artículos Seleccionados: 1

Fig. 3. Measures selection with aggregation function.

semantics to the dimensions. Therefore, in the right panel we display all the paths that lead to the selected dimension so that the user can disambiguate. In the case of *Country* there is no ambiguity. However, for *Fuel* the user select the path *Fuel_type*.



□	Dimensiones
☑	Country
□	Energy_Company
□	EUMember
□	Europe
☑	Fuel
□	NorthAmerica
□	OECDMemeber
□	US_State

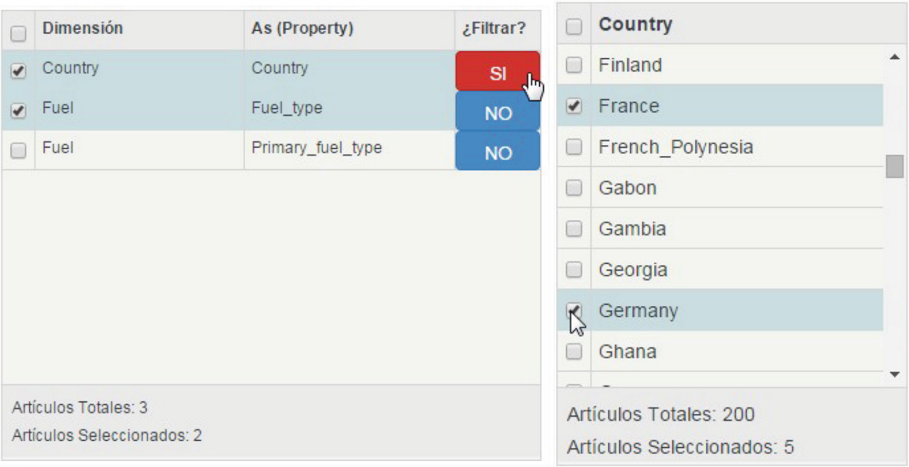
Artículos Totales: 13
Artículos Seleccionados: 2

□	Dimensión	As (Property)	¿Filtrar?
☑	Country	Country	NO
☑	Fuel	Fuel_type	NO
□	Fuel	Primary_fuel_type	NO

Artículos Totales: 3
Artículos Seleccionados: 2

Fig. 4. Dimensions selection and optional filtering.

The prototype also offers the feature of dimension filtering. This is shown in Fig. 5 and can be enabled with the button next to each dimension. In case we activate this option, a new panel will appear (i.e., right panel) displaying the individuals that belong to the selected dimension. In this case, the user wants



□	Dimensión	As (Property)	¿Filtrar?
☑	Country	Country	SI
☑	Fuel	Fuel_type	NO
□	Fuel	Primary_fuel_type	NO

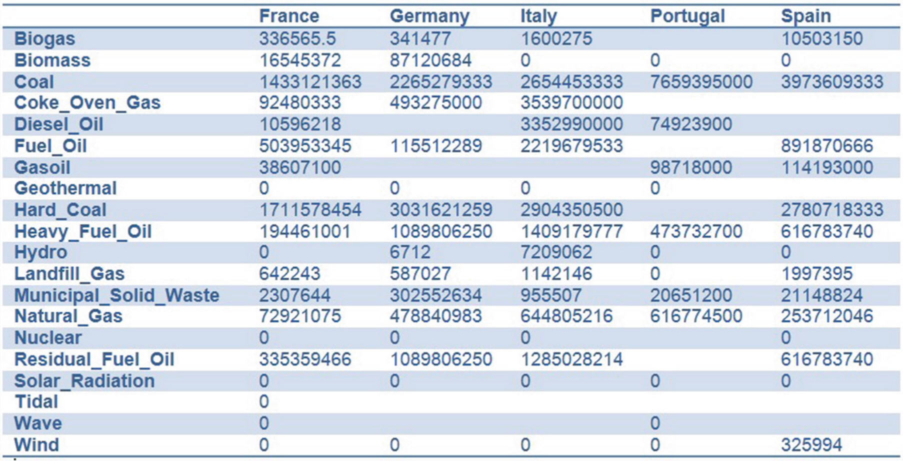
Artículos Totales: 3
Artículos Seleccionados: 2

□	Country
□	Finland
☑	France
□	French_Polynesia
□	Gabon
□	Gambia
□	Georgia
☑	Germany
□	Ghana

Artículos Totales: 200
Artículos Seleccionados: 5

Fig. 5. Filter dimensions by value.

to filter the values of the dimension *Country* and selects only specific countries for the analysis.



	France	Germany	Italy	Portugal	Spain
Biogas	336565.5	341477	1600275		10503150
Biomass	16545372	87120684	0	0	0
Coal	1433121363	2265279333	2654453333	7659395000	3973609333
Coke_Oven_Gas	92480333	493275000	3539700000		
Diesel_Oil	10596218		3352990000	74923900	
Fuel_Oil	503953345	115512289	2219679533		891870666
Gasoil	38607100			98718000	114193000
Geothermal	0	0	0	0	
Hard_Coal	1711578454	3031621259	2904350500		2780718333
Heavy_Fuel_Oil	194461001	1089806250	1409179777	473732700	616783740
Hydro	0	6712	7209062	0	0
Landfill_Gas	642243	587027	1142146	0	1997395
Municipal_Solid_Waste	2307644	302552634	955507	20651200	21148824
Natural_Gas	72921075	478840983	644805216	616774500	253712046
Nuclear	0	0	0		0
Residual_Fuel_Oil	335359466	1089806250	1285028214		616783740
Solar_Radiation	0	0	0	0	0
Tidal	0				
Wave	0			0	
Wind	0	0	0	0	325994

Fig. 6. MD query results.

Finally, Fig. 6 shows the results of the MD query. These can be displayed in different formats and using different visualizations or exported to a file.

7 Conclusions and Future Work

In this paper we have presented the foundations and a prototype tool for LD analysis and visualization following a MD approach. We have proposed MD analytical stars as the foundation to enable the user to easily query LD sets. The MD patterns (i.e., dimensions and measures) suggested by the web-based developed tool to create the stars are based on the semantics of the data (i.e., they provide a conceptual summary of the data), follow the MD model (i.e., information is modeled in terms of analysis dimensions and measures) and are extracted following a statistical approach. The tool has been implemented using web-based technologies.

As future work, it would be interesting to group semantically similar dimensions into categories to offer a cleaner view to the user. Another interesting functionality is to provide dimension hierarchies to be able to perform the classical roll-up/drill-down operations of OLAP over the results. In a near future we will also address the current issue of having a batch process for calculating the dimensions and measures off-line. We aim to apply query sampling methods directly over the LD endpoint to be able to dinamically calculate dimensions and measures. This process will be smoothly integrated in the web-based tool.

Acknowledgements. The work has been partially funded by the CICYT project TIN2011-24147 from the Spanish Ministry of Economy and Competitiveness (MINECO). We would like to thank Iván Posilio for his truly involvement in the development of the prototype.

References

1. Alzogbi, A., Lausen, G.: Similar structures inside RDF-graphs. In: LDOW. CEUR Workshop Proceedings, vol. 996 (2013)
2. Araújo, S., Schwabe, D.: Explorator: a tool for exploring RDF data through direct manipulation. In: LDOW. CEUR Workshop Proceedings, vol. 538 (2009)
3. Auer, S., Lehmann, J.: What have Innsbruck and Leipzig in common? Extracting semantics from Wiki content. In: Franconi, E., Kifer, M., May, W. (eds.) ESWC 2007. LNCS, vol. 4519, pp. 503–517. Springer, Heidelberg (2007)
4. Berners-Lee, T., Chen, Y., Chilton, L., Connolly, D., Dhanaraj, R., Hollenbach, J., Lerer, A., Sheets, D.: Tabulator: exploring and analyzing linked data on the semantic Web. In: Proceedings of the 3rd International Semantic Web User Interaction (2006)
5. Dadzie, A., Rowe, M.: Approaches to visualising linked data: a survey. Semant. Web **2**(2), 89–124 (2011)
6. Etcheverry, L., Vaisman, A.A.: Enhancing OLAP analysis with Web cubes. In: Simperl, E., Cimiano, P., Polleres, A., Corcho, O., Presutti, V. (eds.) ESWC 2012. LNCS, vol. 7295, pp. 469–483. Springer, Heidelberg (2012)
7. Heath, T., Bizer, C.: Linked Data: Evolving the Web into a Global Data Space. Synthesis Lectures on the Semantic Web. Morgan & Claypool Publishers, San Rafael (2011)
8. Heim, P., Lohmann, S., Stegemann, T.: Interactive relationship discovery via the semantic Web. In: Aroyo, L., Antoniou, G., Hyvönen, E., ten Teije, A., Stuckenschmidt, H., Cabral, L., Tudorache, T. (eds.) ESWC 2010, Part I. LNCS, vol. 6088, pp. 303–317. Springer, Heidelberg (2010)

9. Kämpgen, B., Harth, A.: Transforming statistical linked data for use in OLAP systems. In: I-SEMANTICS 2011, Graz, Austria, 7–9 September 2011. ACM International Conference on Proceedings Series, pp. 33–40 (2011)
10. Khatchadourian, S., Consens, M.P.: ExpLOD: summary-based exploration of interlinking and RDF usage in the linked open data cloud. In: Aroyo, L., Antoniou, G., Hyvönen, E., ten Teije, A., Stuckenschmidt, H., Cabral, L., Tudorache, T. (eds.) ESWC 2010, Part II. LNCS, vol. 6089, pp. 272–287. Springer, Heidelberg (2010)
11. Kimball, R., Ross, M.: The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling. Wiley, New York (2011)
12. Klímek, J., Helmich, J., Nečaský, M.: *Payola*: collaborative linked data analysis and visualization framework. In: Cimiano, P., Fernández, M., Lopez, V., Schlobach, S., Völker, J. (eds.) ESWC 2013. LNCS, vol. 7955, pp. 147–151. Springer, Heidelberg (2013)
13. Klyne, G., Carroll, J.J.: Resource description framework (RDF): concepts and abstract syntax (2004). http://www.w3.org/TR/rdf-concepts
14. Nebot, V., Berlanga, R.: Building data warehouses with semantic Web data. Decis. Support Syst. **52**(4), 853–868 (2012)
15. Nebot, V., Berlanga, R.: Towards analytical MD stars from linked data. In: Proceedings of the International Conference on Knowledge Discovery and Information Retrieval (KDIR 2014), KDIR 2014, pp. 117–125 (2014)
16. Prudhommeaux, E., Seaborne, A.: SPARQL query language for RDF (2008). http://www.w3.org/TR/rdf-sparql-query
17. Schraefel, M.C., Shadbolt, N.R., Gibbins, N., Harris, S., Glaser, H.: CS AKTive space: representing computer science in the semantic Web. In: WWW, pp. 384–392. ACM, New York (2004)
18. Stadler, C., Lehmann, J., Höffner, K., Auer, S.: LinkedGeoData: a core for a Web of spatial open data. Semant. Web J. **3**(4), 333–354 (2012)
19. Zviedris, M., Barzdins, G.: ViziQuer: a tool to explore and query SPARQL endpoints. In: Antoniou, G., Grobelnik, M., Simperl, E., Parsia, B., Plexousakis, D., De Leenheer, P., Pan, J. (eds.) ESWC 2011, Part II. LNCS, vol. 6644, pp. 441–445. Springer, Heidelberg (2011)