Discovery and Visual Analysis of Linked Data for Humans

Vedran Sabol[1,2], Gerwald Tschinkel[1], Eduardo Veas[1], Patrick Hoefler[1], Belgin Mutlu[1], and Michael Granitzer[3]

[1] Know-Center vsabol|gtschinkel|eveas|phoefler|bmutlu@know-center.at
[2] Graz University of Technology
[3] University of Passau Michael.Granitzer@uni-passau.de

Abstract. Linked Data has grown to become one of the largest available knowledge bases. Unfortunately, this wealth of data remains inaccessible to those without in-depth knowledge of semantic technologies. We describe a toolchain enabling users without semantic technology background to explore and visually analyse Linked Data. We demonstrate its applicability in scenarios involving data from the Linked Open Data Cloud, and research data extracted from scientific publications. Our focus is on the Web-based front-end consisting of querying and visualisation tools. The performed usability evaluations unveil mainly positive results confirming that the Query Wizard simplifies searching, refining and transforming Linked Data and, in particular, that people using the Visualisation Wizard quickly learn to perform interactive analysis tasks on the resulting Linked Data sets. In making Linked Data analysis effectively accessible to the general public, our tool has been integrated in a number of live services where people use it to analyse, discover and discuss facts with Linked Data.

1 Introduction

The already huge amount of valuable information available in the Linked Open Data (LOD) cloud keeps growing at increasing rate. Unfortunately, this wealth of openly available data is difficult to access and analyse. Without having in-depth knowledge on semantic technologies, such as SPARQL, this abundance of information remains inaccessible. The fact that Linked Data (LD) by definition exhibits a graph structure, even when it describes numeric facts, further complicates the situation. Graph structures, being inherently complex to evaluate and interpret, are not what the majority of users are accustomed working with.

Our goal is to empower users without semantic technology background to search, explore and analyse LD. We strive to make LD accessible to the general public, enabling them to utilise the knowledge stored therein. To do so, we developed tools and workflows designed to be as easy as possible to use. The complexities imposed by semantic technologies and by the LD format are hidden from the user, while at the same time we exploit the advantages arising from semantically rich data. Two web-based interfaces highlight our toolchain: the

Query Wizard [6] and the Visualisation Wizard (Vis Wizard)[12]. Query Wizard makes searching in LD as simple as with standard web search engines, and provides a tabular interface supporting transformations on the retrieved data set (e.g., selecting/removing columns, filtering and aggregation). The Vis Wizard automatically derives visualisations of the created data sets and supports their interactive analysis using multiple coordinated visualisations.

A major novelty introduced by this paper is in realising integrated end-to-end workflows bringing extraction, search, transformation and interactive analysis of LD to "non-experts". While tools addressing each task separately have been previously described and evaluated in isolation, to our knowledge no single system or combination of tools has been reported enabling either experts or non-experts to accomplish these tasks in integrated workflows. We also present results of a formative usability evaluation focusing on visual analysis of LD, and discuss lessons learnt from deploying the workflows.

We draw the motivation and elicit requirements along two exemplary scenarios: i) discovery and analysis of LOD, and ii) analysis of research data embedded in scientific publications (described in Section 3). These scenarios drive the evolution of our tools, developed in the CODE[4] EU project, which went public and are live since late 2012. Design decisions and implementation are detailed in Section 4. Section 5, elaborates on the services that deploy our tools, and illustrates the scenarios (i) and (ii) in practical use cases. We present results of a formative usability evaluation (Section 6), and discuss benefits and lessons learnt along the design, development, and deployment of our tools.

2 Related Work

The problem of easy-to-use interfaces for accessing LD is still largely unsolved. The majority of current tools do not target regular web users. For example, Sindice [17], a major Semantic Web search engine, is practically useless for ordinary web users due to its complex user interface. Freebase Parallax [7] featured the ability to browse sets of related things, and was one of the few web-based tools that provided a table view for results. Both Freebase Parallax and the Falcons Explorer [3] featured a search box as the main entry point, a central idea in our prototype. Yet, in both cases the table view was not the central focus. OpenRefine[5] (formerly Google Refine) supports RDF and there are extensions such as LODRefine[6] that focus on LD. But its main goal is on cleaning tabular data and, although the interface is browser-based, it is not available as a web service. Our work has similarities with faceted search and navigation as described in [13] or [5], and used in OpenRefine, SIMILE Exhibit [8] or DBpedia's instance of Virtuoso's Faceted Search & Find feature[7]. Query Wizard further incorporates interactive elements and concepts from spreadsheet applications.

[4] CODE Project Website: `http://code-research.eu`
[5] `http://openrefine.org`
[6] `http://code.zemanta.com/sparkica`
[7] `http://dbpedia.org/fct`

Stolte et. al proposes a table-based interface for data in relational databases [16]. The system automatically suggests visualisations and coordinates the interaction between them. The mapping of data onto visual properties of a visualisation is not performed automatically, but has to be formulated by the user. Vispedia [2] is a web-based system to create visualisations for articles in Wikipedia. It is limited to Wikipedia data and requires users to choose one of the available visualisations and formulate the mappings manually. Many Eyes [18] is a public web site to upload, visualise and share visualisations. Its data model is a raw table similar to CSV (Comma-separated Values). It uses heuristics to determine whether a column is numeric or text, but it does not automate visualisation. CubeViz [14], similar to the Vis Wizard, enables visualisation and visual querying of statistical RDF Data Cubes. In contrast to our approach, it does not automatically suggest possible visualisations, neither does it support data cubes with multiple measures nor varying number of dimensions. The framework does not rely on semantic description of charts and offers a comparably restricted number of chart types. In [1] a method for automatic mapping of data attributes to visual attributes is described, but no automatic selection of visualisations for a given data set is supported. Marcello et. al[11] confirms the problem the semantic community is currently facing when trying to bring LOD search results in a way that users are comfortable with.

3 Scenarios

We begin by defining two usage scenarios, and then derive the central requirements for the proposed web-based toolchain.

Scenario 1 – Search and Analysis of Linked Open Data is our main scenario which focuses on the openly available information in the LOD cloud. A well-known example is open governmental data, such as made available by EU Open Data Portal[8], which provides a wide variety of statistical facts on our society. The capability to search for and analyse such data would benefit both the general public as well as professionals (e.g. data journalists). Therefore, our Scenario 1 shall consist of the following steps:

1. **Searching** for information in the LOD cloud.
2. **Transforming and preparing** the discovered data for analysis.
3. **Visualising and analysing** the resulting data set to generate new insights.

Using conventional means, the first two steps can be achieved by formulating and executing complex SPARQL queries against an endpoint. Obviously, users without knowledge of semantic technologies will need a simpler solution than that. Concerning the visualisation step, graph visualisation is usually employed because the information is provided as RDF. Instead, employing visualisations suitable for the particular type of information (e.g. statistical, temporal, geographical etc.) would significantly aid the interpretation of data.

[8] EU Open Data Portal: `http://open-data.europa.eu`

Scenario 2 – Analysing Scientific Publication Data addresses another source of hard to utilise, high-quality information: research data present in tables which are embedded in scientific publications in PDF format. In order to access and analyse such data one first needs to extract the tabular information from the PDF. The extracted tables, which typically contain numeric information, shall be semantically described in order to facilitate further analysis. Therefore, our Scenario 2 shall consists of the following steps:

1. **Extracting research data** present in tables embedded in PDF files.
2. **Visualisation and Analysis** of the extracted data set.

With common tools, the first step is achieved by copy-pasting from the PDF and transcribing the table back into the tabular form. Using a spreadsheet application users can correct the data and move it to the correct table cells, which is a laborious task. Visualisation is supported by spreadsheet applications, although users must manually select and configure the charts. Obviously, it would be beneficial for users if major parts of this process were automated.

3.1 Requirements

Taking into account the targeted user group and the defined scenarios, we derive a set of high-level requirements our toolchain needs to fulfil. In the following we differentiate between non-functional (NFR) and functional requirements (FR):

- **NFR1 - Ease of Use**: Tools targeting the general public should be as easy to use as possible. We shall, wherever possible, make use of UI concepts a typical user is already acquainted with.
- **NFR2 - Automation**: The tools should maximise automation and eliminate unnecessary steps which currently must be performed by the user.
- **NFR3 - Exploiting Semantics**: The system should exploit Semantic Web standards and the semantics of the data to the advantage of the user.
- **FR1 - LOD Search**: A search tool shall support retrieval in SPARQL endpoints.
- **FR2 - Data Transformation**: A tool shall support transforming and refining of the found/extracted data set.
- **FR3 - PDF Table Extraction**: A tool shall provide extraction of tabular data from scientific publications in PDF format.
- **FR4 - Data Triplification**: A triplification tool shall provide functionality for exporting of a data set as RDF.
- **FR5 - Interactive Visualisation**: A visualisation tool shall support visualisation and interactive analysis of Linked Data sets.

4 Proof of Concept

Driven by requirements NFR1 – NFR3, we present central design decisions for the tools introduced by FR1 – FR5. After that, as a proof of concept, we introduce workflows showing how these tools are employed to realise the scenarios defined in the previous section.

4.1 Design Decisions

Searching in LD shall come as close as possible to what users are accustomed with the major search engines (e.g. Google). The entry point to search shall be a search box that works as what is expected for standard (non-semantic) web search. After performing a full-text search the returned results shall be presented in the form of a table, where a row corresponds to a single subject, a column represents a predicate, and cells contain objects for the given subject and predicate. The rationale behind using tabular representation is that users are familiar with tables and are often proficient in using spreadsheet applications. Also, the tabular form is suitable for refining and transforming the retrieved data set. For example, with just a few clicks user shall be able to add and remove columns (i.e. predicates), filter the results (rows) depending on simple criteria, and aggregate (group by) columns. A tool supporting the described functionality, the **Query Wizard**, satisfies requirements NFR1, NFR3, FR1 and FR2.

Extracting tabular data from scientific publications and exporting it as RDF shall be the task of the **Data Extractor** tool. We choose the W3Cs RDF Data Cube Vocabulary[9] [4] which provides a semantic framework for expressing multi-dimensional data sets as Linked Data. The Data Extractor is composed of three components: i) an embedded **PDF Extractor** which takes a PDF file as input and returns extracted tables as output [10], ii) an (optional) user interface for correcting extraction errors and defining dimensions and measures of the data cube, and iii) a triplifier which exports a tabular data set as RDF Data Cube. Importantly, tabular data sets created by the Query Wizard already contain semantic information, which is utilised by the triplifier to create RDF Data Cubes in a fully automatic mode. The Data Extractor, with the embedded PDF Extractor, satisfies the requirements NFR2, NFR3, FR3 and FR4.

Semantic information present in the RDF Data Cubes shall be utilised to enable automated visualisation. Depending on semantic data characteristics, automatic visualisation suggests meaningful visual representations and disables those not suitable for the data. The automatism also includes configuring a visualisation, i.e. mapping different columns of the data set onto suitable visual properties (e.g. axes, colours etc.) of the visualisation. When multiple visualisations and configurations are possible, the user shall have the freedom to select only between the meaningful ones. Also, for complex multi-dimensional data, it should be possible to generate multiple visualisations (e.g. a geo- and a time-visualisation) in order to provide insights into different aspects of the data set. A tool supporting the described functionality, called the **Visualisation Wizard** (Vis Wizard for short), satisfies requirements NFR1, NFR2, NFR3, and FR5.

With this we have defined the design of a set of tools which satisfy the requirements derived in the previous section. Next, we briefly outline workflows specifying how these tools are employed to implement the scenarios.

[9] RDF Data Cube Vocabulary: `http://w3.org/TR/vocab-data-cube`

4.2 Workflows

Workflows shown in Figure 1 describe how the proposed tools are employed to realise the two scenarios.

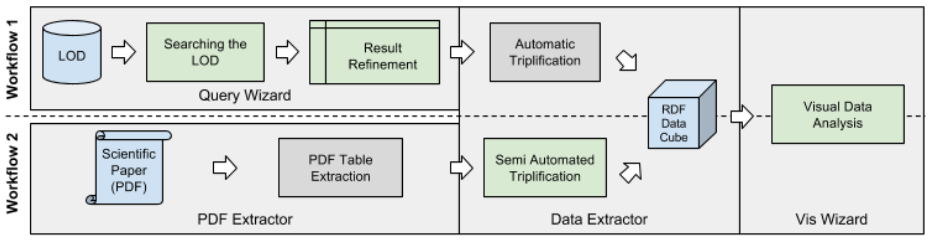


Fig. 1. Workflows using the proposed tools to implement the scenarios.

Workflow for Scenario 1 (up in the Figure 1) implements the process of searching the LOD, interactively transforming the data, automatically converting it into a data cube, and generating interactive visualisation of the data set. The Query Wizard, which accesses the Linked Open Data cloud, is used to execute full-text searches on an endpoint. Query Wizard is also employed for the next step: displaying the retrieved data in tabular form and manipulating it (e.g. selecting columns, filtering or aggregating). Following that, the data extractor automatically generates an RDF Data Cube relying on semantic information in the data. For the last step the Vis Wizard is used to automatically generate visualisations to support interactive analysis of the data.

Workflow for Scenario 2 (bellow in the Figure 1) implements the process of extracting tabular data from publications, converting the data into an RDF data cube, and generating interactive visualisation of the data. In the first step the PDF Extractor analyses the structure of a scientific paper in PDF format and automatically extracts tables. In the next step, the extracted tabular data is passed to the Data Extractor which provides a user interface for semi-automatic triplification. Data Extractor analyses the data and suggests dimensions and measures of the cube. The user can perform corrections if the automatic data analysis produced errors. Following that an RDF Data Cube is generated and can be stored into a Linked Data endpoint. In the last step the user can analyse the data using the Vis Wizard. Optionally, before visualising, Query Wizard can be applied to display and transform the data set.

4.3 Selected Implementation Details

In this section we briefly describe the most relevant technical solutions behind our tools. For detailed reading the corresponding publications should be consulted.

Query Wizard: The Query Wizard [6] turns search terms entered by the user into a series of SPARQL queries, which are then sent to the endpoint chosen by the user. First, with the help of a full-text index, a search in all the rdfs:labels is performed, and the first 10 matching subjects are returned. Search results are shown as a table, where a row corresponds to a single subject and a column represents a predicate (by default the first row displays rdfs:label, the second rdf:type). Cells contain objects, i.e. any number of literals and/or entities, depending on the row and column. Using the SPARQL 1.1 COUNT feature the total number of matching results is also retrieved. Another query is generated to display all available predicates for the displayed subjects. When the user selects one of these predicates from the drop-down list, another query is performed to retrieve the respective data. When users set a filter on one of the displayed columns, a whole new set of SPARQL queries is generated and sent to the endpoint. With just a few interactions, the system can produce hundreds of lines of SPARQL code – all completely invisible to the user. Also, thanks to the aggregation features of SPARQL 1.1, tasks that usually involve a Pivot Table or specialised Data Warehousing software – such as calculating averages, sums, minima, maxima, or counts based on selected dimensions – can be performed by the Query Wizard with the help of a simple interface. The Query Wizard can also be used to explore available RDF Data Cubes which are publicly available through a SPARQL endpoint. The front page features automatically generated lists of RDF Data Cubes for endpoints such as EU Open Data[10] or Vienna Linked Open Data[11]). However, support for selecting endpoints based on available data is not included, representing an opportunity for future research.

Data Extractor and PDF Extractor: The Data Extractor [15] uses a semantically enriched HTML table produced by the Query Wizard to guess dimensions and measures of a data cube. The columns of the table are automatically classified as either dimensions (if the cell content is non-numeric), measures (for numeric cell content), or multi-value (if there are multiple values in at least one of the cells of the given column). For extracting tabular information from publications the embedded PDF extractor [10] analyses the structure of a PDF document using unsupervised machine learning techniques and heuristics. Contiguous text blocks and geometrical relations between them are extracted from the character stream. The blocks are categorised into different classes resulting in a logical structure of the document. Table extraction starts from a "table" caption, and then labels neighbouring sparse blocks recursively as table blocks, if their vertical distance is within a specific threshold.

Visualisation Wizard: To suggest appropriate visualisation for a data set in a RDF Data Cube, we developed visualisation vocabulary, which describes visualisations semantically in an OWL ontology. The vocabulary describes: (1) visualisations in terms of visual channels $va : hasVisualChannel(va : Chart, va : VisualChannel)$ (e.g., axes, colour, size), and (2) visual channels in terms of data types $va : hasDatatype(va : VisualChannel, va : DataType)$ (e.g., boolean, nu-

[10] `http://open-data.europa.eu`
[11] `http://cweiss.net/lod`

meric). For a particular visualisation a visual channel may be optional, which is represented in $va : Occurrence$. The mapping algorithm identifies valid relations from $qb : dimensions$ to $va : VisualChannels$ in a $va : Chart$. The relation (mapping) between the RDF Data Cube is only valid, when the data types of the cube components and visual channels are compatible[12]. After analysing the data type compatibility the Vis Wizard automatically suggest any of the 10 currently available visualisations and valid mapping combinations.

The Vis Wizard offers interaction facilities to organise, refine and inspect the visualised data with coordinated brushing, mouse-over inspections, filtering and aggregation. Brushing and linking is a powerful interactive analysis technique, which combines different visualisations to overcome the shortcomings of single techniques [9]. Interactive changes made in one visualisation are automatically reflected in the other ones. Vis Wizard utilises semantic information (i.e. dimension URIs) to link different visualisations, which may be displaying different Data Cubes. These are created, for example, when data sets are aggregated.

5 CODE Tools in Use

During the development of the Query and Vis Wizards we followed the "release early, release often" principle. As soon as a new feature was complete and ready for testing, it immediately rolled out to our staging server and, if no major problems were found, a short time later it was publicly available at our production server. The prototypes have been online since November 2012 and have been under permanent scrutiny of fellow researchers and other interested colleagues for a year and a half now. Since then, the Query Wizard alone generated around 100.000 SPARQL queries that users did not have to formulate themselves, whereby this number comprises all queries generated within interactive exploration tasks. Both tools are available under:

- Query Wizard: http://code.know-center.tugraz.at/search
- Visualisation Wizard: http://code.know-center.tugraz.at/vis

Integration with other platforms: The Query Wizard and the Vis Wizard have been integrated into the 42-data[12]. 42-data is a data-centric question and answer platform, which focuses on discussions and answers backed by empirical facts in numerical LOD. Embedded Query Wizard tables and Vis Wizard visualisations facilitate exploration and analysis of LOD sets within the platform. Uptake of the 42-data social platform is steadily increasing usage of our tools. Another integration which benefits the usage rates of our tools is with the commercial MindMeister[13] mind mapping web platform. It enables the Query Wizard to export data sets in the form of mind maps, which can be shared and collaboratively edited by MindMeister users. Also, visualisations generated by the Vis Wizard can be added to mind maps as images which link back to the original interactive charts.

The following two use cases demonstrate the implementation of the scenarios.

[12] 42-data Platform: `http://42-data.org`
[13] MindMeister Mind Mapping platform: `http://www.mindmeister.com/`

5.1 Use Case 1

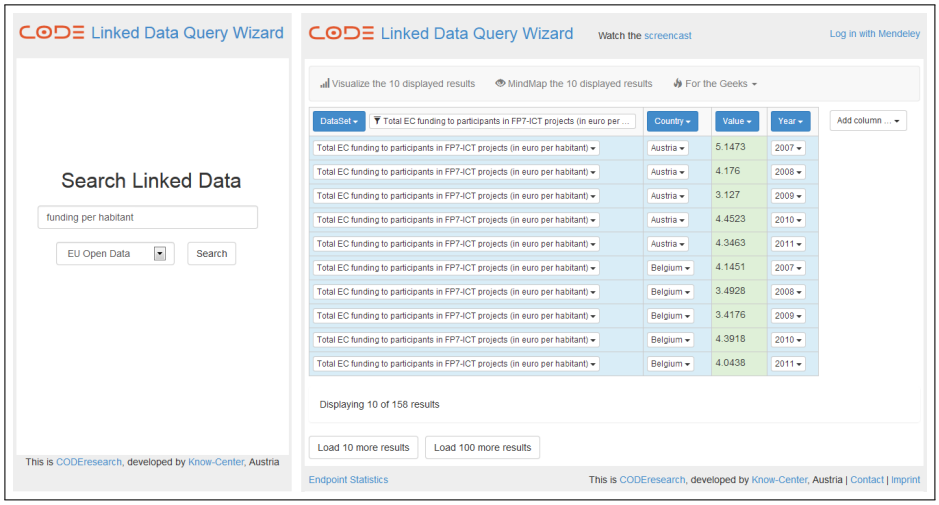


Fig. 2. Query Wizard: Searching (on left), tabular result representation (on right)

Our first and primary use case implements Scenario 1 – Search and Analysis of Linked Open Data. Assume the user is interested in total EU project funding for different countries. Using the Query Wizard he selects an appropriate endpoint, in this case the EU Open Data endpoint, and search for "funding per habitant" (see Figure 2, on left). Search results, displayed in tabular form (Figure 2, on right), can be manipulated through filtering and adding/removing columns. Filtering by data value is performed by clicking on the specific value and selecting filtering in the drop-down list. Columns are added by using the "Add Column" button and selecting a predicate. Columns are removed by clicking on the column header and selecting "Remove column". In the shown data set we added the column "Dataset", filtered it by the value "Total EC funding to participants in FP7-ICT projects (in euro per habitant)", and added the columns "Country", "Year" and "Value" to obtain the data we are interested in. Before visualising we load the whole data set consisting of 158 entries using the "Load more results" buttons.

To visualise the data set we click on "Visualize the displayed results" link which loads the data into the Vis Wizard (see Figure 4). Six out of ten available visualisations are enabled for our data set ("Possible Charts" in the Figure 4). We select the scatterplot (left in the Figure 4) to visualise funding (y-axis) for countries (x-axis) in different years (colour coding). Next, we want to find out how the funding is spread over Europe. To achieve this we aggregate the data for countries by averaging over the years. A simple aggregation dialogue allows us

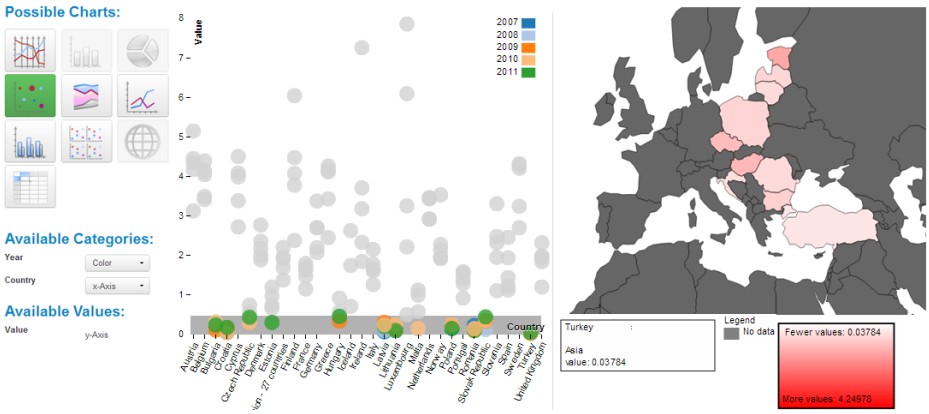


Fig. 3. Vis Wizard showing data on EC funding to participants, per habitant. The left chart displays funding for countries in different years. The right chart shows the average funding by country.

to select "Country" and choose "average" as aggregation function. Aggregation generates a new data set which is visualised in a geo-chart (right chart in the 3), where colour coding is used to visualise the average funding (a deeper shade indicates higher funding). Finally, we want to discover which countries receive the lowest amounts of funding. Due to available semantic information Vis Wizard knows that "Country" has the same meaning in the original and aggregated data set. This allows us to apply a brush in the scatterplot (shown as grey rectangle) selecting the countries with lowest received amounts of funding. Brushing operation greys out all non-selected countries in both visualisations, which leads us to a new insight: in the geo-chart we can clearly observe that countries with the lowest amounts of funding are located in Eastern Europe. A screencast of the use case is available on YouTube[14].

5.2 Use Case 2

The second use case (schematically shown in Figure 4) briefly demonstrates the usage of our tools in Scenario 2 – Analysing Scientific Publication Data. Our user is interested in analysing research data available in a scientific paper. In particular, for the paper "Combined Regression and Ranking" from D. Sculley (2010) the user wishes to explore results found in "Table 1: RCV1 Experimental Results" (on left in the Figure 4). We start by uploading the PDF file into the Data Extractor which internally uses the PDF Extractor to extract the tables. Data Extractor guesses which rows and columns represent data cube dimensions and which cells contain observations. Next, the user selects the first table which is presented in an interface showing dimensions marked in blue and observation

[14] Use case 1 screencast: http://www.youtube.com/watch?v=mA_vi1F7TSE

marked in green (in the middle of the Figure 4). The user has the opportunity to edit the table and, if necessary, correct extraction errors by: marking dimensions, removing columns and rows, modifying cell content etc. When ready, with a single click the table is converted into an RDF Data Cube and visualised in the Vis Wizard (parallel coordinates visualisation shown on right in the Figure 4).

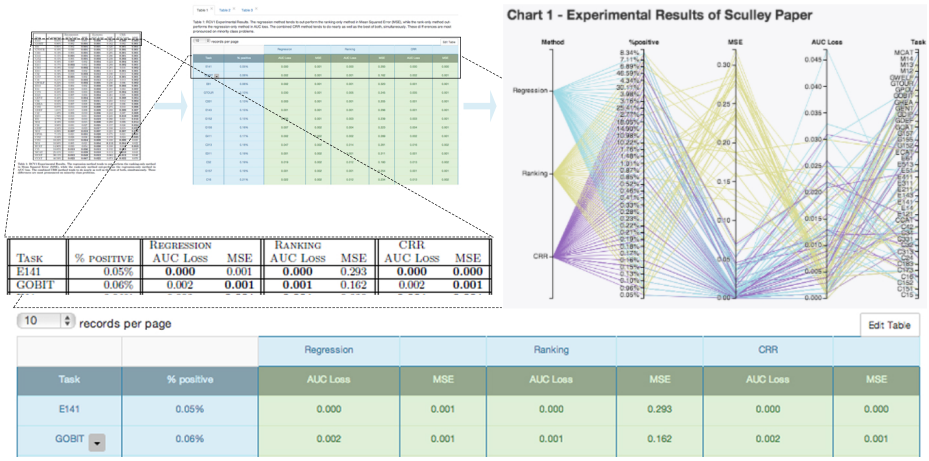


Fig. 4. Use Case 2 begins with a PDF document containing tables (on left), which are extracted and annotated with the RDF Data Cube Vocabulary using the Data Extractor (in the centre), and then visualised with the Vis Wizard (on right).

6 Formative Evaluations

This section presents formative usability evaluations performed with the precise goal to ascertain that users can: a) search, refine and transform LD, b) perform interactive analysis on complex data sets. The evaluations consisted of several tasks that required a conceptual understanding of different inherently complex operations on LD. In both cases we used the standard NASA Task Load Index (TLX) to measure workload in loosely time constrained tasks and followed the Thinking Aloud protocol to uncover usability issues. The time constrain was loosely maintained, meaning the moderator kept track of the timing but was not fully strict. No timer was shown to participants. This is a common way to introduce time pressure that participants need to keep track of mentally while performing the task. The time constrain combined with the Thinking Aloud adds up to effort and frustration when participants cannot progress as expected. Workload was computed with the simplified NASA R-TLX.

6.1 Evaluation 1: Search and Refinement

This section summarises a formative evaluation with eight participants focusing on search, refinement and transformation of LD, initially reported in [6]. The evaluation showed that people could perform these tasks using the provided abstractions (high Performance TLX), with little stress (Mental/Temporal Demand). Interestingly, people quickly learnt new features (mid-low Effort fluctuation). The Query Wizard was generally well received, both by users with and without a computer science background. The main point of critiques was a missing possibility to add URI filters through a menu in the table header. Additional suggestions for improvements were to show the total number of results more prominently and to implement an infinite scroll mechanism that automatically loads more data as the users scroll to the bottom of the screen.

6.2 Evaluation 2: Visual Analytics

The focus of this evaluation was the interactive analysis of complex datasets. 8 people participated in this evaluation (age in range$[24-38]$), all without background in LD or semantic technologies.

Methodology. The evaluation required participants to perform operations such as filtering, aggregation and brushing. After a demographics questionnaire, participants received a short guide to the Query and Vis Wizards, showcasing aforementioned functions but without explaining the meaning or underlying constructs thereto. Then, participants performed 6 tasks as shown in Table 1.

T1: Filtering in the Query Wizard	T4: Aggregation - Multiple Categories
T2: Filtering in the Vis Wizard	T5: Aggregating Multiple Values
T3: Aggregation	T6: Brushing in Multiple Views

Table 1. Tasks. Summary of tasks and corresponding activities in the experiment.

For example, the instruction for T1 was: *please show the data set in Query Wizard. We are interested only in the countries which have a CO2 Emission over 13 Tons per persons. After that, please visualise the results. You have 3 minutes to complete this task.* Upon finishing the task or when time-up was called, participants filled the NASA TLX and a subjective assessment questionnaire. An exit questionnaire was used to collect preferences and suggestions.

Quantitative subjective workload. From 54 tasks performed in total by 8 participants, 39 were successfully completed in time. Results on workload were positively below the $\frac{1}{3}^{rd}$ of the scale. T1 and T5 rated lowest on workload. T1 was the first task that we deemed less complex and received the lowest mental demand (MD) rating ($M = 12.5, Std = 10.35$) accordingly. MD remained stable in subsequent tasks. Temporal demand did not present major differences across

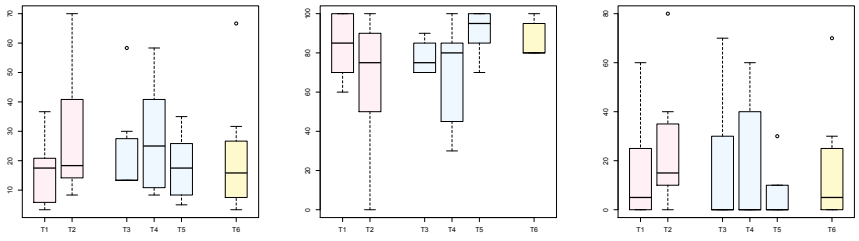
tasks. The main visual analytics tasks T5 and T6 present high perceived performance ratings ($M = 91.25, Std = 11.25$, $M = 86.25, Std = 9.16$), accompanied by relatively low frustration (T5: $M = 6.25, Std = 10.60$).

Qualitative Thinking Aloud. Participants found it difficult to select the proper dataset in the first task (T1), but they clearly understood that they needed to use a filter, set the filter correctly and visualised the data without complications. Participants choose two general strategies to solve T2, either set a filter in the Query Wizard first and show the filtered data (6), or visualise the data and brush the parallel coordinates to filter (2). T3 was solved almost unanimously by visualising and then aggregating data. One participant aggregated the data first and then visualised it. Participants found T4 suddenly complex, mainly because the initially suggested visualisation was not the appropriate one to solve the task, but also by the need to group and aggregate data. Five participants grouped incorrectly at first, and after noting the error, had to redo it. Only two participants solved this task without issues. In T5, participants showed all the skills acquired throughout the experiment, three participants used parallel coordinates and two used scatterplot matrices to solve the task, two other participants used grouping and aggregation. Only one participant had difficulties with multiple aggregated values. In T6, two participants were confused by a usability issue of the brush in a scatterplot, but all could actually solve the task.

	T1		T2		T3		T4		T5		T6	
TLX	M	STD	M	STD	M	STD	M	STD	M	STD	M	STD
Mental Demand	12.5	10.35	33.75	18.46	28.75	26.42	35	26.18	31.25	22.32	30	27.25
Physical Demand	5	7.55	7.5	10.35	10	20.70	17.5	27.64	8.75	13.56	21.25	29.48
Temporal Demand	15	11.95	25	33.80	15	11.95	16.25	17.67	18.75	15.52	15	17.72
Effort	32.5	24.34	43.75	24.45	42.5	27.64	47.5	17.52	33.75	20.65	31.25	25.87
Frustration	15	22.67	25	25.63	16.25	26.15	17.5	25.49	6.25	10.60	16.25	24.45
Performance	83.75	15.97	66.25	34.20	77.5	8.86	68.75	25.31	91.25	11.25	86.25	9.16
Workload	16.04	11.01	28.125	21.01	22.5	15.88	27.5	18.89	17.91	10.71	21.25	20.50

Table 2. Workload of Visual Analytics. Results on workload from the VA experiment. Green tones show positively lowest ratings, and red tones the opposite higher ratings.

Preferences and exit questionnaires. Users liked the workflow for data analysis, and were in general motivated to discover facts in the data through the provided functionality. They found it fast, simple and intuitive to use, and regarded the design highly. They appreciated the automated suggestion and mapping of visualisations. Still, participants rated the colours in the geo-chart really poor, and were at times confused by the brushing functionality. To the question, *what would you use this toolset for?*, participants answered: to visualise any kind of statistical data, server log analysis, project tracking and to quickly answer questions which involve data analysis. Finally, when asked if they could have solved the tasks with other tools of their choice, which ones they would have used, participants unanimously replied they would search with Google, and manually collect and copy the data into a spreadsheet application for analysis.



(a) Workload (b) Performance (c) Frustration

Fig. 5. Workload from R-TLX computed inverse performance. Colours encode the type of task (filtering: pink, aggregation: blue, multiple views: yellow).

6.3 Discussion and Lessons Learnt

The study was designed as formative and not comparative in nature, to discover if people could actually perform otherwise complex analytics operations on LD. Thus instead of seeking statistical deviation from a baseline, we opted for finding correlation between observations, the Thinking Aloud, and TLX results. The formative evaluation sheds light on the expressive power of our toolset for interactive analysis of LD. Participants could apply complex operations with minimal effort on large datasets. The TLX provides interesting results. Although tasks were constrained in time, participants did not feel pressed to finish (TD in table 2). They had more difficulties to solve T2 and T4, observable in the higher variance in results and confirmed in the Thinking Aloud. We hypothesise that this effect is due to confusing paths to the solution, either in the form of different strategies (T2) or because the Vis Wizard did not open the right tool for the task. Yet, people were confident solving tasks that involved more complex operations (performance in T5 and T6, Figure 5(a)). Indeed, participants expressed notably less effort and frustration to achieve higher performance in these tasks. One notable result was that people could convert narrative description of the task to a set of operations on the data without having to cope or knowing the complexity of the underlying implementation of these operations. In the following we summarise lessons learnt and directions for future work gained throughout the process of development, testing and deployment.

Data volume ignored. People seemed oblivious to the volumes of data they were actually handling. In the Query Wizard experiments people ignored the loaded number of records in the data set in several occasions, requiring time to realise that more records were available. Visualisations helped stress the issue, as visual abstractions may leave people without a clue as to the size of the data.

Unrefined suggestion of visualisations. Visualisations are suggested based on the characteristics of the data, whereby multiple visualisation may meaningfully represent a data set. However, many users were confused if the presented visualisation would not help solve the particular problem at hand. For them,

selecting the right visualisation for a task is not trivial. Ranking of possible visualisations depending on data and context will improve suggestion quality.

Navigating paths to a solution. Most problems can be solved by combining operations in different ways. However, in cases when it became clear than an incorrect path to the solution was taken, participants often had to restart from the beginning. Backtracking in the operations applied to the data is not always supported, and clearly enough, people get frustrated by having to repeat steps.

Analytics workflow. An exploratory process is one of hypothesis, experiment and discovery. To the experienced practitioner these stages are clear, but not so for novices. In this sense, although our tools let novices reach into the richness of LD, the analytics workflow is only implicitly supported. To facilitate the work of both novices and experts, our future tools need to include explicit representations of this workflow, so users can move back and forth along stages.

7 Conclusion

We have deployed a toolset that makes LD accessible to the general public. We used well-known metaphors to ensure a smooth learning of our tools, which hide the underlying technological complexities from users (NFR1), automate the analytical process (NFR2) through automated visualisation and Data Cube extraction, and leverage semantic technologies (NFR3) for both automation and interactive analysis. Our evaluations show that non-experts could pose complex queries and discover facts from LD using interactive visual analysis.

The CODE toolset has been online since well over a year and has been actively used for accessing and analysing Linked Open Data. Both tools are deployed as part of the 42-data Q&A platform, with the purpose of supporting data-centric discussions. Users of the MindMeister service benefit from the capability of our Wizards to generate mind maps from data sets and visualisations. The presented toolset opens a wealth of interesting avenues for research as well as for direct deployment in productive applications. It is our hope that these results will motivate other practitioners and scientists to try and incorporate described workflows and tools in their work, to design better solutions for LOD based on lessons exposed, to integrate end-points, enrich LOD with saved queries processed data to access and utilise LOD analysis in their daily work.

Acknowledgements. This work is funded by the EC 7th Framework project CODE (grant 296150). The Know-Center GmbH is funded by Austrian Federal Government within the Austrian COMET Program, managed by the Austrian Research Promotion Agency (FFG).

Query Wizard and Vis Wizard were developed by Know-Center's Knowledge Visualisation team. We thank our colleagues from the Knowledge Discovery team for providing the PDF Extractor, and our colleagues from the University of Passau for providing the Data Extractor.

References

1. Cammarano, M., Dong, X.L., Chan, B., Klingner, J., Talbot, J., Halevey, A., Hanrahan, P.: Visualization of heterogeneous data. In: IEEE Information Visualization

2. Chan, B., Wu, L., Talbot, J., Cammarano, M., Hanrahan, P.: Vispedia: Interactive visual exploration of wikipedia data via search-based integration. IEEE Trans. Visualization & Comp. Graphics (Proc. InfoVis) (2008)
3. Cheng, G., Wu, H., Gong, S., Ge, W., Qu, Y.: Falcons Explorer: Tabular and Relational End-user Programming for the Web of Data. In: Semantic Web Challenge (2010)
4. Cyganiak, R., Reynolds, D.: The RDF Data Cube Vocabulary (2013)
5. Erling, O.: Faceted Views over Large-Scale Linked Data. Linked Data on the Web (LDOW) (2009)
6. Hoefler, P., Granitzer, M., Veas, E., Seifert, C.: Linked Data Query Wizard: A Novel Interface for Accessing SPARQL Endpoints. In: Proceedings of Linked Data on the Web (LDOW) at WWW (2014)
7. Huynh, D., Karger, D.: Parallax and companion: Set-based browsing for the data web. WWW Conference (2009)
8. Huynh, D.F., Karger, D.R., Miller, R.C.: Exhibit: lightweight structured data publishing. Proc. of the 16th int. conf. on World Wide Web Banff, Alb (2007)
9. Keim, D.A.: Information visualization and visual data mining. IEEE Transactions on Visualization and Computer Graphics (2002)
10. Klampfl, S., Kern, R.: An unsupervised machine learning approach to body text and table of contents extraction from digital scientific articles. In: Aalberg, T., Papatheodorou, C., Dobreva, M., Tsakonas, G., Farrugia, C.J. (eds.) TPDL. Lecture Notes in Computer Science, vol. 8092, pp. 144–155. Springer (2013)
11. Leida, M., Afzal, A., Majeed, B.: Outlines for dynamic visualization of semantic web data. In: Meersman, R., Dillon, T.S., Herrero, P. (eds.) OTM Workshops. Lecture Notes in Computer Science, vol. 6428, pp. 170–179. Springer (2010)
12. Mutlu, B., Höfler, P., Tschinkel, G., Veas, E.E., Sabol, V., Stegmaier, F., Granitzer, M.: Suggesting visualisations for published data. In: Proceedings of IVAPP 2014. pp. 267–275 (2014)
13. Oren, E., Delbru, R., Decker, S.: Extending faceted navigation for RDF data. In: The Semantic Web ISWC 2006 5th International Semantic Web Conference ISWC 2006 Athens GA USA November 59 2006 Proceedings. vol. 4273 (2006)
14. Salas, P.E., Martin, M., Mota, F.M.D., Breitman, K., Auer, S., Casanova, M.A.: Publishing statistical data on the web. In: Proceedings of 6th International IEEE Conference on Semantic Computing. IEEE 2012, IEEE (2012)
15. Seifert, C., Granitzer, M., Hoefler, P., Mutlu, B., Sabol, V., Schlegel, K., Bayerl, S., Stegmaier, F., Zwicklbauer, S., Kern, R.: Crowdsourcing fact extraction from scientific literature. In: Workshop on Human-Computer Interaction and Knowledge Discovery (SouthCHI). LNCS, vol. 7947. Springer (2013)
16. Stolte, C., Hanrahan, P.: Polaris: A system for query, analysis and visualization of multi-dimensional relational databases. IEEE Transactions on Visualization and Computer Graphics 8, 52–65 (2002)
17. Tummarello, G., Delbru, R., Oren, E.: Sindice.com: Weaving the open linked data. In: Aberer, K., Choi, K.S., Noy, N.F., Allemang, D., Lee, K.I., Nixon, L.J.B., Golbeck, J., Mika, P., Maynard, D., Mizoguchi, R., Schreiber, G., Cudré-Mauroux, P. (eds.) ISWC/ASWC. Lecture Notes in Computer Science, vol. 4825, pp. 552–565. Springer (2007)
18. Viegas, F.B., Wattenberg, M., van Ham, F., Kriss, J., McKeon, M.: Manyeyes: a site for visualization at internet scale. IEEE Transactions on Visualization and Computer Graphics (2007)