Suggesting Visualisations for Published Data

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

[1]*Know-Center, Graz, Austria*

[2]*University of Passau, Germany*

[3]*Graz University of Technology, Austria*

{*bmutlu, phoefler, gtschinkel, eveas, vsabol*}*@know-center.at, stegmaier@dimis.fmi.uni-passau.de, michael.granitzer@uni-passau.de*

Keywords: Linked Data; RDF Data Cube; Visualisation; Visual Mapping; Research Data

Abstract: Research papers are published in various digital libraries, which deploy their own meta-models and technologies to manage, query, and analyze scientific facts therein. Commonly they only consider the meta-data provided with each article, but not the contents. Hence, reaching into the contents of publications is inherently a tedious task. On top of that, scientific data within publications are hardcoded in a fixed format (e.g. tables). So, even if one manages to get a glimpse of the data published in digital libraries, it is close to impossible to carry out any analysis on them other than what was intended by the authors. More effective querying and analysis methods are required to better understand scientific facts. In this paper, we present the web-based CODE Visualisation Wizard, which provides visual analysis of scientific facts with emphasis on automating the visualisation process, and present an experiment of its application. We also present the entire analytical process and the corresponding tool chain, including components for extraction of scientific data from publications, an easy to use user interface for querying RDF knowledge bases, and a tool for semantic annotation of scientific data sets.

1 INTRODUCTION

We are currently confronted with a continuous, massive increase of published content, and the applied methods used by current digital libraries are not sufficient anymore. They mainly expose the research knowledge using domain-specific meta-models and technologies, such as the widely used Dublin Core meta-model (Powell et al., 2005). But these meta-models mostly focus on structural attributes like title, author or keywords, and rarely consider the content of the publications. Hereby, the ability to effectively find the desired information is limited due to weakly-defined querying attributes. In addition, scientific data or facts included in publications are unstructured or, at best, in tabular format, so that once the information is found, there is hardly a way to reuse it.

Our goal is to provide a tool to automatically extract data from scientific publications and propose the appropriate means to visualise the facts and data therein. Figure 1 illustrates the envisioned workflow with a scenario, starting with extraction of data from a publication to visualisation. The main concern of this paper is the automated suggestion of visualisations appropriate for the data contained in a publication. Furthermore, we strictly focus on suggesting only proper visual tools (e.g. visualisations that really apply to the data), avoiding failure cases that render the analysis process tedious. To do so, the CODE Visualisation Wizard[1] (Vis Wizard) (Mutlu et al., 2013) relies on a prior extraction and organization of the unstructured, scientific data in a publication into Linked Open Data (LOD) by using the RDF Data Cube Vocabulary[2]. The strength of LOD lies in its interlinking structured data in a format that can be read and processed by computers. The Vis Wizard then applies semantic technologies to derive and propose visual analysis tools.

In order to automate the process of generating and proposing visual analysis tools, it is necessary to integrate visualisation aspects into the Semantic Web. Our contribution is, on the one hand, a vocabulary, which describes the semantics of visualisations and of their mapping to the corresponding data. On the other hand, we introduce the process that drives the automated visualisation workflow. After introducing relevant work in Section 2, we summarise the complete process in Section 3, carried out in the frame of

[1]CODE Visualisation Wizard: `http://code.know-center.tugraz.at/vis`

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

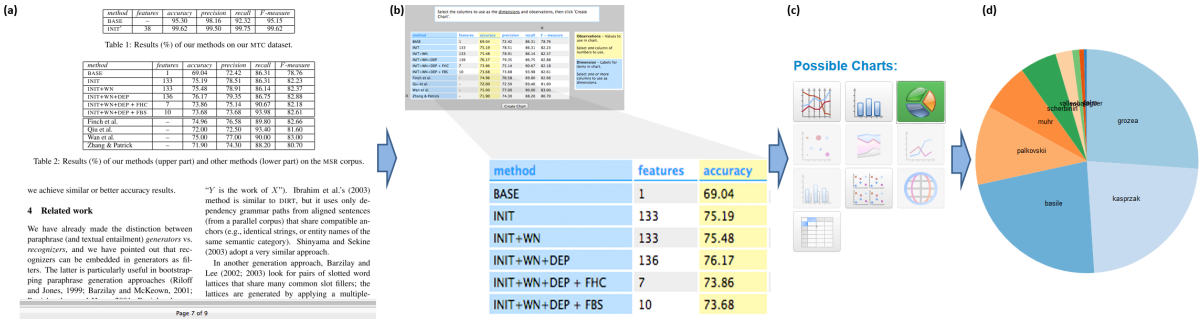

Figure 1: Visual analysis of published data: one scenario of application for automated extraction and visualisation of data. (a) a pdf with a bunch of tables containing interesting data. (b) the UI to structure the extracted data from the pdf (c) visualisations suggested for the current data, grayed-out options cannot be reified. (d) visual analysis UI allows to reassign mappings of visual channels (e.g. axes, colors, sizes etc.) to data.

the EU funded CODE project (Stegmaier et al., 2012) . Thereafter, in Section 4, we concentrate on the main topic, our approach to automatically generating visualisations. In Section 5 we give a brief summary of its evaluation, and finally we conclude and outline some important topics for the ongoing work in Section 7.

2 RELATED WORK

In this section, we overview of most relevant approaches with focus on methods for describing and analyzing visualisations in general. We also review related approaches on visualisation of structured data.

2.1 Semantics of Visualisations

The general method to describe and structure the Linked Data is called the Resource Description Framework[3] (RDF). Semantic description of visualisations using RDF is a new research topic and the literature, up to now, offers just a few related works. The most significant research (Dumontier et al., 2010) comes from the biomedical domain, called the Statistical Graph Ontology (SGO), and presents a new approach to manage statistical graph knowledge by semantic annotation of graphs. Describing graphs semantically has been recognized as a very effective way for querying and analyzing research data visually (Voigt et al., 2013). Because of the tight coupling between visualisations and statistical data achieved in this way, the querying is considerably simplified and the data can be directly mapped onto visualisations.

While SGO provides a sophisticated ground for describing statistical graphs, some attribute are missing in this ontology to enable automated visualisation like the attribute datatype, instantiation occurrence, etc. (more details about this in Section 4.1). These attributes may be useful to build a model that defines the mapping between data and visualisations.

2.2 Visualization and Analysis of Papers

Close to our concept of analysing scientific facts contained in research publications, Utopia Documents (Attwood et al., 2010) developed a PDF reader to simplify searches for relevant information. This intelligent tool provides additional data (links to suitable web resources and metadata) and methods to interactively analyse the content of the evaluation results contained in papers (in table to be exact). For the visualisation, the table content and figures are enriched with provenance information. Visualizations are created following a very common principle: columns are statically mapped to axes of a scatterplot. There is no vocabulary defining the structure of this data and, therefore, no intermediate model which would allow to make more sophisticated visualization mappings.

2.3 Visualisation of RDF Data Cubes

An approach which is very similar to our Vis Wizard is realized within the scope of the CubeViz Framework (Salas et al., 2012). This framework supports the visualisation and visual querying for statistical data sets, stored in form of the RDF Data Cube. However, in contrast to our approach, there is no support for automated suggestions of possible visualisations and also no support for large cubes (i.e. cubes having varying number of dimensions and multiple measures). Furthermore, the framework does not rely on semantic description of charts and it offers just few very ordinary chart types.

[3] RDF: www.w3.org/RDF/

3 WORKFLOW OVERVIEW

3.1 The CODE Platform

The CODE project[4] offers a platform to structure research data and release them as Linked Data (Bizer et al., 2009). Linked Data describes methods to publish and to interlink structured data (meta-data) on the World Wide Web. The intent of these methods is to connect data with semantic technologies, making them automatically readable by computers. In the CODE project, Linked Data act as a basis to publish and to interlink research data (Seifert et al., 2013), thereby strongly focusing on their content and not only on structural attributes.

Figure 1 shows the CODE approach to organize and analyze research publications. To bring published data to the hands of the user in an intuitive manner, CODE envisions a workflow with the following major steps: (a) Data Extraction, (b) Integration and Aggregation, and (c) Analysis and (d)Visualisation. Automation of this workflow is essential for analysts who have to integrate huge amounts of research knowledge in short time. For instance, the first two steps deal with the automated extraction and integration of the research knowledge into a common meta-model (in further text, vocabulary - e.g. from the unstructured text stored in the PDF format, to structured LOD in RDF), while the last step offers the automated support for visualising that knowledge. Section 4 looks further at the process to suggest visualisations.

3.2 Visualisation Workflow

Consider the following scenario: While looking at a publication, Jane feels overwhelmed with numbers spread across tables throughout the pages. To make sense of these data, she quickly *exports* it to the CODE visual analysis tools. Before visualizing the data, Jane has to specify the dimensions and measures (in further text, RDF Data Cube Components) of the data. The Vis Wizard then *suggests* appropriate visualisations, which Jane can then fine-tune to her liking.

Figure 1 shows how the above scenario is realised with CODE components. The first step, *extracting* the data, is automated by the CODE PDF Extractor (Klampfl et al., 2013) , which extracts tables from scientific publications. In the second step, *exporting* the data in an appropriate format the CODE Data Extractor (Schlegel et al., 2013) is used to semantically annotate the table (i.e. specify dimensions and measures, and their types), producing an RDF Data Cube. We chose the RDF Data Cube Vocabulary (RDF-DCV) because it was developed by the W3C to represent statistical data (e.g. the research results from tables in a publication) (Salas et al., 2012).

Once a data set is available in the RDF Data Cube format it is passed to the Vis Wizard. In this third step, *mapping* the data onto visualisations, a mapping algorithm uses the semantic descriptions of visual components and the semantic annotations of the data to suggest visualisations, suitable for that particular data set. The user only needs to choose a visualisations by pressing one of the enabled buttons and the chosen visualisation will automatically generated and displayed. The fourth step, *visualising* the data set, allows the user to modify the mapping of visual channels (i.e. visual attributes such as axes, size or colour of visual items, etc.) to the structured data. The user has the option to re-adjust how the data columns are mapped to the visual channels, whereby only meaningful mappings are permitted. It is also possible to generate additional visualisations for the same data set, which are displayed within the same browser window, empowering the user to analyse different aspects of a heterogeneous data set in a combined view.

3.3 The Data Representation

The RDF-DCV represents data as a collection of so called *observations*, each consisting of a set of *dimensions* and *measures*. Dimensions identify the observation, measures are related to concrete values. For example: in the dataset for the PAN[5] scientific challenge, that evaluates software for uncovering plagiarism developed by different teams. The RDF-DCV includes a collection of observations with dimensions describing the *teams* and with concrete values for the *challenge result* (Figure 2 shows a sample visualisation).

Therefore, using the RDF-DCV, one such observation is created for each of the statistical values in a publication. The format guarantees a uniform representation for all (unstructured) statistics, thereby enabling the Vis Wizard to access data in a standard way defined by the RDF Data Cube specification.

4 AUTOMATED VISUALISATIONS

Once the data are extracted, the Vis Wizard undertakes the complex task of suggesting only the appro-

[4] CODE: code-research.eu/

[5] PAN: http://pan.webis.de/

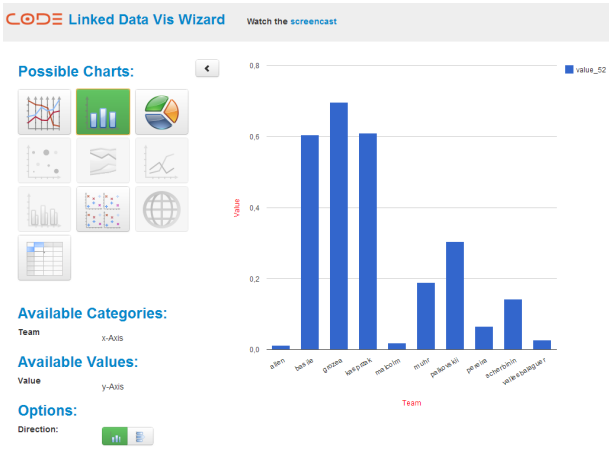

Figure 2: The automatically generated visualisation of PAN Data with *Team* as dimension and with a *Challenge Result* as measure.

priate visualisations. To do so, it relies on semantic descriptions of visualisations in terms of visual channels and mappings, supplemented with a process for semantic mapping.

Figure 3 shows the layered architecture of the process to automate the suggestion of visualisations. Similar to the common vocabulary used in CODE to structure heterogeneous data as Linked Data, we define a common vocabulary to describe and integrate visualisations in the aforementioned CODE workflow. In the following, we describe involved vocabularies and their relevance to research data and the visualisation process.

4.1 Describing Visualisations

To support the process of mapping visualisations to data described in RDF-DCV, we developed a Visual Analytics Vocabulary (VA Vocabulary[6]) that describes visualisations semantically as an OWL[7] ontology. Our semantic description strictly focuses on describing the visual encoding process, hence we represent visualisations in terms of their visual channels (Bertin, 1983). However, instead of pursuing a thorough specification encompassing all known facts about visual perception as (Voigt et al., 2012), we concentrate on pragmatic, simple facts that will aid the sensible mapping (e.g. (Mackinlay, 1986)), extending the description to many different types of visualisations. Thus, we have separated our VA Vocabulary into two parts: (1) the model of an abstract visualisation (i.e. an abstract visualisation type) that captures only the commonalities shared between all concrete visualisations, and (2) concrete visualisation models, which capture just specific information. Concrete visualisations refine the abstract visualisation model depending on their type. The abstract visualisation model specifies most important structural components that any concrete visualisation may have. These are:

- **Name:** Identification for a visualisation.
- **Visual Channel:** A container for data, which have to be visualised. It contains structural rules required to correctly map statistical data to visualisation. For example, a visual channel for a bar chart is refined to represent its `x-axis` and `y-axis`.
- **Description:** A collection of non-mandatory components (e.g. textual description or image such as SVG figure for a concrete visualisation).

The difference between concrete visualisations lies in their reification of visual channels. For example, a bar chart has only two visual channels, `x-axis` and `y-axis`. According to their type definition, `y-axis` always represents a numeric (e.g. decimal, float, integer, etc.), whereby `x-axis` supports more types (e.g., categorical, string). Further, this visualization will be suggested only if both visual channels are provided and have data (we say here, they are instantiated). In this case, both visual channels are mandatory for the bar chart. In contrast, parallel coordinates, require at least one `x-axis` and additional instances of that visual channel are optional, a characteristic shared by tabular visualisation and the scatterplot matrix.

To capture such differences in our VA Vocabulary, we characterize visual channels with the following attributes:

- **Datatype:** Defines a set of primitive datatypes that a visual channel can support.
- **Occurrence:** Defines the cardinality of a visual channel (i.e. how many instances are allowed for the concrete visual channel).
- **Persistence:** Defines whether a visual channel is mandatory part of the concrete visualisation or not.

The occurrence attribute identifies whether a visual channel can be instantiated only once (e.g. bar chart `x-axis` and `y-axis`, see Figure 2) or multiple times (e.g. parallel coordinates `x-axis`). There are two different values for this attribute: *one* and *many*. The occurrence *many* is used for visualising high-dimensional RDF Data Cubes. In contrast, the occurrence *one* defines a fixed cardinality.

[6] VA Vocabulary: `code-research.eu/ontology/visual-analytics`
[7] Web Ontology Language: `www.w3.org/TR/owl-features/`

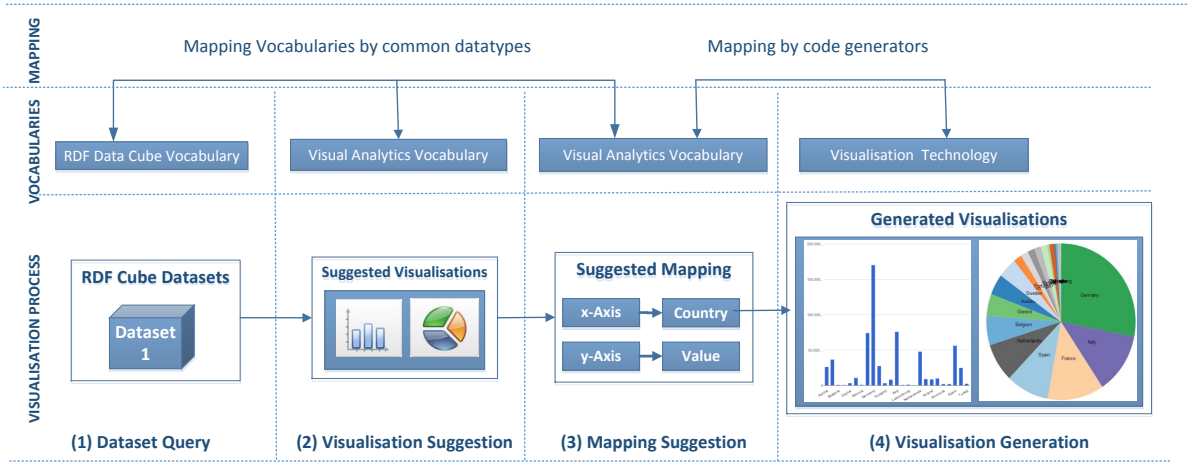

Figure 3: Main parts of the Visualisation Wizard: automated visualisation process (bottom), vocabularies (middle) and mapping vocabularies (top).

The persistence attribute helps define more complex cases. For example, a visualisation with three mandatory and two optional visual channels. Hereby, the case with the parallel coordinates can be alternatively defined as follows: one mandatory visual channel with the occurrence *one*, and another one, which has an occurrence *many* and *no* persistence.

4.2 Suggesting Visualisations

The mapping between both mentioned vocabularies, the RDF Data Cube and the VA Vocabulary, is a relation from dimensions and measures in the former to the corresponding visual channels of a visualisation in the latter. This relation is valid only if the datatypes of the cube components and visual channels are compatible. Datatype compatibility in our context means having exactly the same primitive datatypes, both conforming to the XSD datatype definitions[8]. Beyond datatype compatibility, a valid mapping needs to account for structural compatibility, since visualisations from the VA Vocabulary may have fixed or varying number of visual channels. To clarify this, let us consider the bar chart from the Figure 2. It has two visual channels, `x-axis` and `y-axis`, and can visualise data only if both channels are instantiated. That means, it can plot the RDF Data Cubes with exactly one dimension and one measure. The additional requirement is that both visual channels support datatypes, which are compatible to datatypes of the RDF Data Cube model.

Visualisations with optional visual channels support different structures of the RDF Data Cube model (i.e. number of dimensions and measures). From these observations, we derive the following requirements for a valid mapping:

- **Structural Compatibility:** The instantiation of dimensions and measures in the RDF Data Cube is unbounded. That is, we can define observations with arbitrary dimensions and measures. Therefore, possible instantiation patterns (i.e. in the format dimension:measure) for each observation are: (1) `1:1`, (2) `1:n`, (3) `n:1` and (4) `n:n`. In order to find a valid mapping, we have to find in the VA Vocabulary visualisations with the same instantiation patterns. Not all visualisations have the same structural definition as a RDF Data Cube; therefore not all visualisations are able to display arbitrary Cubes. Thus, only those visualisations with the instantiation patterns that match the observations of the RDF Data Cube are candidates for the valid mapping. This is called structural compatibility in our context.
- **Datatype compatibility:** The structural compatibility is not sufficient to claim the correctness of the mapping. Therefore, in addition to the structural compatibility, the visual channels and related RDF Data Cube components have to be compatible regarding their datatypes.

If for a given RDF Cube model at least one visualisation does match both requirements above, there is a valid mapping and the RDF Data Cube can be visualised. The following pseudo-code shows the mapping algorithm.

The attributes of visual channels form the basis to prove both types of the compatibility. For a given RDF Data Cube, the algorithm returns either a list of visualisation candidates and concrete mappings between data dimensions, measures and visual channels,

[8] XSD Datatypes: `www.w3.org/TR/2001/REC-xmlschema-2-20010502/`

```
Data: RDF Data Cube
Result: set(mapping suggestions)
get visualisation candidates;
get observation components;
while visualisation candidates exist do
    instantiate visual channels;
    generate all combinations for datatypes of
    visual channels;
    while datatype combinations exist do
        map combination to instantiated visual
        channels;
        pack mapping configuration;
        if ( ((occurrence matches) and
        (persistence matches)) and (type
        matches) ) then
            add to mapping suggestion set;
        else
            throw invalid mapping
            configuration;
```

Algorithm 1: Simplified algorithm for determining feasible mapping suggestions

or nothing. According to the algorithm above, we first get all concrete visualisations from the repository and observation components from the RDF Data Cube. For each visualisation candidate, we instantiate its visual channels according to the structure of the observations (i.e. according to the number of existing dimensions and measures within an observation). This means, we produce the structure, which is similar to the one of the observations. Hereby, we achieve the structural compatibility between both models, but it needs to be proved. In the next step, we generate all combinations of the form *visual channel* $\rightarrow$ *datatype* and pack those combinations as a mapping configuration. The mapping configurations are candidates for a valid mapping. In the last step, we verify each of these mapping configurations. First, the structural compatibility is verified by checking the occurrence and persistence attributes of the visual channels. Based on these attributes, we identify whether a visual channel is missing or not allowed. The mapping configurations which pass this step, satisfy the structural compatibility requirement and are forwarded to type checking, where datatypes of the visual channels are compared with the datatypes of the observations. Finally, only those mapping configurations, which pass this last verification step, are considered valid suggestions.

5 PRELIMINARY EVALUATION

We performed a preliminary evaluation, principally to identify usability gaps, but also to explore the reaction to recommended visualisations as well as the mapping suggestions. Instead of following the complete workflow from extraction to visualisation, we concentrated on the latter part. Thus we used datasets that had been previously structured as RDF-DC. This also insured the independence from other services and components of the CODE workflow, offering a more controlled evaluation environment.

During the experiment, participants were mainly exposed to the Query Wizard and the Vis Wizard. The Query Wizard (Höfler et al., 2013) aims to help users select relevant data from Linked Data repositories, and is one of the components for visual analysis inside of the CODE platform. The user only performs a keyword search in the CODE Semantic Web endpoints and gets the resulting data presented in an easy-to-use web-based interface.

5.1 Procedure

The evaluation procedure started with a demonstration of the Vis Wizard. Three different example datasets were visualized, each with incremental complexity: one dataset with one dimension and one measure, one dataset with two dimensions and one measure, and the last one with three dimensions and one measure. During the demonstration the user received a description on how a visualisation would be suggested. We also introduced the fact that the Vis Wizard proposes a visualisation with a default mapping, and described how to set a new mapping.

After this demonstration, users were presented with one of ten datasets randomly chosen from the Query Wizard to visualize it and answered evaluation questions based on a simple analysis task. Every one of these datasets collects data from the European Union (EU), referring different statistics to funding amounts per country. The datasets were constrained to two dimensions and one measure (country, year, and funding, respectively), which corresponds with our mid-level complexity in the initial demonstrator. The task was chosen to let the participants get a feeling of using the interfaces to obtain data from these massive repositories. Thus, the task was to analyze the funding amounts distributed over countries and find the country that was assigned the largest amount in 2010.

The next task was to figure out the funding received by countries in 2010, in ascending order, from lowest to largest. This simple task had particular im-

Figure 4: The result of the second Task of the evaluation.

plications, since it forced participants to interact with the mappings for a given visualisation to figure out the results.

5.2 Participants

The heuristic evaluation was performed by ten IT experts: 8 males and 2 females. Some of the participants were experienced in the visualisation of Linked Data whereas the others had little or no experience in this area. As our Vis Wizard suggests all possible visualisations for a given dataset, and participants were free to choose different ones, we did not collect any quantitative measures. We did, however collect subjective feedback towards the overall usage of the Vis Wizard, and the appreciation of interacting with mappings.

5.3 Results

The results of this preliminary evaluation are as follows:

- All participants found the Vis Wizard easy to use after a short introduction. They especially liked the aesthetic of the website, according to their opinion there is neither too much nor too little information, buttons or icons on the website.
- The user perception was very good concerning the limitations on selecting invalid mappings, since the Vis Wizard only allowed the selection of suggested and valid visualisation combinations.
- The collection of the charts was sufficient for all users.
- The first mapping done by the server was not always satisfactory by the users. However, the ability to easily set a new mapping variant changed this.
- Sometimes there was too much data which has been visualized so that the identification of the data was difficult. This is the reason why the user prefers to have the option to take only a part the data in order to have a clear visual representation.
- The user also wanted to have the option to zoom, to filter and to select the data on the visualisations.

During the first task of finding the country which received the largest funding in 2010, users interacted for example with parallel coordinates or scatterplot matrix. For the second task, one example solution was to select a time based visualisation and organize the mappings so that year was on the `x-axis`. As a result, country data is distributed across the other axis (see Figure 4).

From our preliminary evaluation, we argue that automating the visualisations for statistical data can be very beneficial for target user group (i.e. researchers, students, etc.). In order to cover many query scenarios, it is necessary to complement the visualisation-based approach with the traditional query such as a tabular one, as shown in this evaluation.

6 DISCUSSION

We have described a full analytical process, going from unstructured data in publications, through extraction and structuring of the data, to visual analysis. We also leverage the wealth of information present in the Linked Open Data Cloud, by making it easily searchable and accessible for visual analysis. We covered a tool-chain instantiating every stage of this workflow. It is available through our website or through tools integrated by our partners (e.g. Mendeley desktop[9]). The motivation was to visualize scientific data from publications. However, our tools are not constrained to the scientific domain, they can also be deployed in any other domain that requires extracting data from published text, such as governmental

[9] Mendeley: `www.mendeley.com/`

reports. Many institutions require manuals for maintenance and finding the threshold numbers, for example, for a calibration procedure, is always a tedious task. The application scenarios for the technology we propose in this paper span numerous areas, both scientific and industrial. Although we have numerous datasets from published Open Data (e.g., EU Open Data), ad-hoc analysis of arbitrary publications is limited by the data extraction process (precision 79%, recall 76%, on ICDAR dataset[10]). For visual analysis, the Vis Wizard can combine visualizations in a single view, but it cannot yet suggest sensible combinations, and interaction across views is limited.

7 CONCLUSION

Organizing and analyzing research publications using current technologies of digital libraries remains a tedious task. The continuous increase of the published content drives a need to find more effective solutions to manage that content.

In this paper, we have outlined and instrumented a workflow, whereby the research data has to traverse several stages, starting from the original and unstructured text to its final structured form and visualisation. The essential aspect of the this approach is the automated support for this workflow. Automating the visualisations allows users to easily find and to analyze research data. In this context, we have developed a common vocabulary for defining the visualisations semantically. Further, in order to identify the matching visualisations for given research data, we have defined a mapping of this vocabulary to the existing vocabulary of that data. Based on these vocabularies and their mapping, we are able to automatically suggest visualisations.

The development of the Vis Wizard will continue throughout the rest of the year and includes the ongoing topic, visualisation refinement. In further the user should have the possibility to aggregate, to filter and to select the data for the visualisations. These refinements will depend on the visualisation features to serve users with intelligent processing options.

ACKNOWLEDGEMENTS

This work is partially funded by the EC 7th Framework projects CODE (grant 296150) and EEXCESS (grant 600601). The Know-Center GmbH is funded within the Austrian COMET Program Competence Centers for Excellent Technologies of the Austrian Federal Ministry of Transport, Innovation and Technology, the Austrian Federal Ministry of Economy, Family and Youth and by the State of Styria. COMET is managed by the Austrian Research Promotion Agency (FFG).

[10] ICDAR: dag.cvc.uab.es/icdar2013competition/

REFERENCES

Attwood, T. K. et al. (2010). Utopia documents: linking scholarly literature with research data. *Bioinformatics*, 26(18).

Bertin, J. (1983). *Semiology of graphics*. University of Wisconsin Press.

Bizer, C. et al. (2009). Linked data - the story so far. *Int. J. Semantic Web Inf. Syst.*, 5(3):1–22.

Dumontier, M. et al. (2010). Modeling and querying graphical representations of statistical data. *Web Semant.*, 8(2-3):241–254.

Höfler, P. et al. (2013). Linked data query wizard: A tabular interface for the semantic web. In *ESWC (Satellite Events)*, pages 173–177.

Klampfl, S. et al. (2013). An unsupervised machine learning approach to body text and table of contents extraction from digital scientific articles. In *International Conference on Theory and Practice of Digital Libraries 2013*, Valetta, Malta.

Mackinlay, J. (1986). Automating the design of graphical presentations of relational information. *ACM Trans. Graph.*, 5(2):110–141.

Mutlu, B. et al. (2013). Automated visualization support for linked research data. In *I-Semantics 2013*.

Powell, A. et al. (2005). Dublin core metadata initiative - abstract model. White paper, Eduserv Foundation, UK, KMR Group, CID, NADA, KTH, Sweden, DCMI.

Salas, P. et al. (2012). Publishing statistical data on the web. In *Semantic Computing (ICSC), 2012 IEEE Sixth International Conference on*, pages 285–292.

Schlegel, K. et al. (2013). Trusted facts: Triplifying primary research data enriched with provenance infoamtion. In *ESWC 2013*.

Seifert, C. et al. (2013). Crowdsourcing fact extraction from scientific literature. In *Workshop on Human-Computer Interaction and Knowledge Discovery*, Maribor, Slovenia. Springer.

Stegmaier, F. et al. (2012). Unleashing semantics of research data. In *Second Workshop on Big Data Benchmarking*, Pune, India.

Voigt, M. et al. (2012). Context-aware recommendation of visulization components. In *The Fourth International Conference on Information, Process, and Knowldege Management*.

Voigt, M. et al. (2013). Capturing and reusing empirical visualization knowledge. In *1st International Workshop on User-Adaptive Visualization*.