SEO4OLAP – Search Engine Optimized Presentation of Statistical Linked Data

Daniel Breucker[1], Benedikt Kämpgen[1] and Andreas Harth[2]

Abstract: Statistical data is published online by a variety of organizations, such as *Eurostat*, the statistical office of the European Union. Some of these datasets are available as Linked Data which has great potential for future research or commercial exploitation. Yet, search engines are struggling to index statistical Linked Data and therefore, normal web users have difficulties to access it. In this paper, we present the approach *SEO4OLAP*, which generates search engine optimized webpages for every possible view on statistical datasets. Statistical datasets are usually based on a Multidimensional Model (MDM) and can therefore be queried by means of Online Analytical Processing (OLAP). We present a new OLAP query model which allows queries to be represented by clean URIs. We also present a mathematical model to compute the overall number of possible views of a dataset. We evaluate our approach by publishing two datasets from *Eurostat* with our *SEO4OLAP* system. We observe that our webpages are indexed by search engines and conclude that the system can have benefits for data publishers, web users and search engine providers.

Keywords: Search Engine Optimization (SEO); Online Analytical Processing (OLAP); Statistical Linked Data

1 Introduction

Statistical data is published online by a variety of organizations, among them public agencies and governmental institutions. The statistical office of the European Union *Eurostat* is one of these institutions. Eurostat publishes statistics such as about trade, population, economy and finance inside the EU. The data can be accessed by web services in the SDMX format, downloaded directly or explored by an online pivot table.

Recent work ([KOH12], [Sa12]) has focused on leveraging statistical data on the Web by means of Semantic Web technologies in conjunction with Online Analytical Processing (OLAP). This has been fostered by the W3C, since they recommended the RDF Data Cube Vocabulary (QB) for modelling multidimensional data (especially SDMX) in the Resource Description Framework (RDF).

Based on the current trend, we assume that in the near future organizations are going to publish statistical Linked Data directly. For the moment, Linked Data wrappers exist for some of those statistical datasets. In the case of *Eurostat*, the corresponding Linked Data

[1] FZI Research Center for Information Technology, Haid-und-Neu-Str. 10-14, 76131 Karlsruhe, breucker@fzi.de, kaempgen@fzi.de
[2] Institute AIFB, Karlsruhe Institute of Technology, Englerstr. 11, 76131 Karlsruhe, harth@kit.edu

wrapper is *Estatwrap* by *OntologyCentral*[3].

Linked Data technologies and the RDF Data Cube Vocabulary allow to publish statistical data in a standardized format, interlink different datasets and retrieve information with the standard query language SPARQL. This has great potential for many different fields. Empirical experiments could be published and reused by other research parties. New services on open data may evolve, leading to new business opportunities.

Despite the potential of statistical Linked Data, human friendly interfaces to explore and interact with it are still suspect to ongoing research ([Ho13], [Mu14]). The SPARQL query language is the most common way to query Linked Data. Since SPARQL can only be used by experts in the field of semantic technologies, the information published in the Semantic Web is not available for usual web surfers.

A common way to retrieve information from the Web is the usage of search engines. Since search engines are focusing on HTML-based content, Linked Data is usually not included in search results. Neither are SDMX representations and views created using JavaScript pages.

This leads to the following problem: Even though statistical data is published online, search engines are not able to retrieve corresponding results. In this paper, we analyse how statistical Linked Data can be published to allow search engines to properly index statistical datasets. We present our approach *SEO4OLAP* which generates search engine optimized (SEO) webpages for all possible facts of a dataset. Once these landing pages are published, search engines should be able to crawl the content. From this approach the following research questions can be derived and will be answered:

- Is SEO4OLAP improving the status quo?
- What is the computation complexity of this problem? How many webpages are created depending on the number of dimensions, measures and dimension members?
- How can semantic mark-up technologies such as *Schema.org*[4] contribute to this?

The remainder of this invited[5] paper is structured as follows. Section 2 presents our approach *SEO4OLAP*, including both the abstract design of the system and a mathematical model to compute the total number of possible views. In section 3, the approach is evaluated. We implemented a concrete system in Java and published two datasets from *Eurostat*. We compare how search engines rank our pages in comparison to the original data source from *Eurostat*. After a discussion of our findings in section 4, we present related work in section 5. Finally, a conclusion with a discussion on future work is presented in section 6.

[3] http://estatwrap.ontologycentral.com/
[4] http://schema.org/
[5] This is an invited contribution from the BigGIS project for the BDSDST 2016 workshop.

2 Approach – Creating Landing Pages for Cubes

The goal of *SEO4OLAP* is to generate search engine optimized webpages for every possible view of a statistical linked dataset modelled in the RDF Data Cube Vocabulary. The schematic process of *SEO4OLAP* is illustrated in Figure 1. The system receives a dataset as input, analyses the data cube schema, computes all possible OLAP-queries and generates webpages for every result with corresponding keywords.

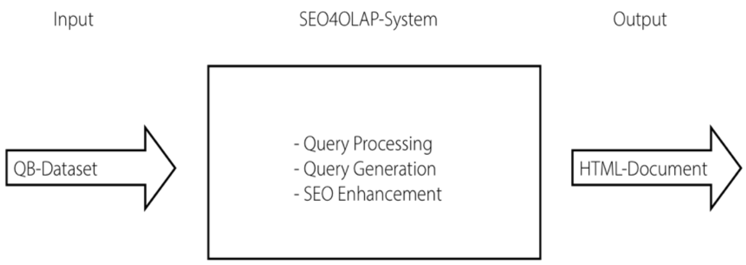

Fig. 1: Schematic process of *SEO4OLAP*

Facts from data cubes can be retrieved through OLAP-queries. Since these facts are represented as a webpage, queries are performed via HTTP GET requests. Therefore, we developed a new OLAP query model which can be represented by pretty URLs.

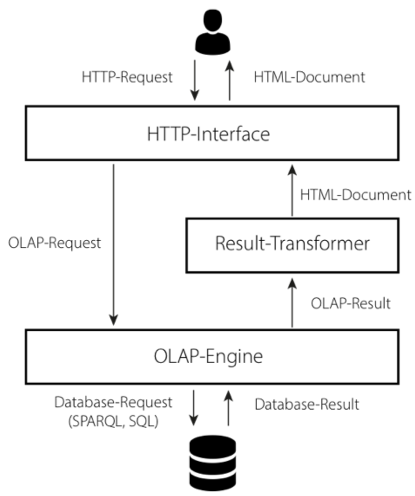

Fig. 2: Query processing of *SEO4OLAP*

The query processing workflow is illustrated by Figure 2. Incoming OLAP-queries are transformed into a series of OLAP-operations, a so called *Logical OLAP Query Plan*. The plan is executed by an OLAP-engine which transforms the OLAP-operations into native database queries, e.g. using SPARQL. Database results are then enhanced with corresponding keywords, e.g. the labels of the selected measure and the diced dimension members. The result is then returned as an HTML page.

The URLs of the facts have to be made accessible to humans and crawlers in order to be found by search engines. This can be achieved by applying a link structure which allows to reach every fact of the dataset. We recommend to add links from fact-webpages to near neighbours. A near neighbour is an OLAP-query with one different parameter. Another possibility to help search engines crawling all webpages is to provide a sitemap. A sitemap is a list of all URLs of a website.

There are various ways to define OLAP-queries and express them with query languages. As an example, MDX allows to specifically define how OLAP-results should be displayed in a pivot table. For our use case, this is too complex. Search engines prefer clean and readable URLs for their ranking. Therefore, we developed a new query model which is presented in the following. First, the parameters are described and afterwards used in our URL-scheme.

We adapt the concept of subcube queries from Kämpgen et al. [KOH12] and define it as per Definition 1.

Definition 1 (Subcube Query): *A subcube query on a certain cube is represented as a tuple (Measures2Project, Dimensions2Keep, Members2Dice), with Measures2Project consisting of identifiers of measures to be projected, Dimensions2Keep consisting of identifiers of dimensions to be kept and Members2Dice consisting of identifiers of members to be diced. Dimensions that are not represented by a member in Members2Dice or part of Dimensions2Keep, are sliced.*

We propose this model because we believe it is a good fit for our use case of SEO. The main advantage is that instead of all dimensions, only relevant parameters have to be set. This means that the query directly defines which measures, dimensions and members should be retrieved. In comparison to Kämpgen et al. [KOH12], less parameters are needed.

As an example, consider a query on an employment dataset asking for the absolute employment number and the employment rate of Germany per year, disregarding the gender. A subcube query for this request would be the following: `({employment absolute, employment rate}, {date}, {germany})`. We assume, that the dimension *Date* only has members on the level year. The dimension *Gender* is sliced, since neither men or women are defined as *Member2Dice* nor is it part of *Dimensions2Keep*.

A subcube query as defined per Definition 1 can be submitted via HTTP by setting the cube identifier and the three parameters *Measures2Project*, *Dimensions2Keep* and *Members2Dice*. A typical HTTP GET request for our scenario has the following structure:

```
http://baseUri/ep?cube=id1&measure=id2&dimension=id3&member=id4
```

An API based on this scheme would work just fine. The problem is that search engines prefer clean and readable URLs consisting of keywords. Therefore, we developed a URL-scheme which enables the same functionality but has a clean appearance with keywords used as identifiers. It is presented in the following:

```
http://baseUri/cubeId/pattern/id1/id2/...
```

The pattern consists of three digits, defining the number of used parameters per group. The following identifiers are slash-separated and in the order of the pattern. As an example, the pattern 122 means that the first identifier is a measure, the following two are dimensions and the last two are members. In the following some example URLs are presented for better understanding.

- `http://example.org/employment/211/absolute/rate/date/germany`:
  A URL for our previous example.
- `http://example.org/employment/112/rate/date/women/france`:
  A URL asking for the employment rate of women in France per year.
- `http://example.org/population/111/absolute/date/poland`:
  A URL asking for population numbers in Poland per year. This is a different data cube than the previous.

A cube's size and thus the number of possible views is exponentially dependent on the number of dimensions. The maximum number of possible views can be computed with Formula 1. Every dimension has d_i possible members plus 2 extra values: the implicit ALL-member, which is a slice and the implicit Zero-Member, when no value is selected; m is the number of measures, n is the number of dimensions. The meaning of d_i, m and n apply for all formulas in this chapter.

$$MaxViews = m \times \prod_{i=1}^{n}(d_i + 2) \tag{1}$$

Formula 1 underlies the following restrictions:

- Only one measure is displayed per view. It has to be noted that this already reduces the potential overall number of views a lot.
- A dimension can be set to one member, be sliced (All-Member) or not set (Zero-Member). Multiple members of the same dimension are not regarded.
- Implicit aggregated members of higher levels are neglected. Only explicit members are regarded.

Due to the exponential growth of the problem, the computational effort to generate separate webpages for every possible view can be enormous. From a SEO and a user experience perspective, it is questionable whether all possible views are necessary. Therefore, we propose two restrictions to our model:

- The *Dice Dimensionality* (DiceDim) restricts the maximum number of dimensions which can be diced.
- The *Number of free Dimensions* (FreeDim) defines the maximum number of dimensions that are neither diced nor sliced, thus free.

Depending on these two restrictions, the total number of possible views can be computed by Formula 2.

$$Views_{DiceDim,FreeDim} = m \times \sum_{d=0}^{DiceDim} \left(FreeDimFactor_{FreeDim} \times DiceDimFactor \right) \tag{2}$$

with

$$FreeDimFactor_{FreeDim} = \begin{cases} \sum_{s=0}^{FreeDim} \frac{(n-d)!}{s!\,(n-d-s)!} & \text{with } \frac{(n-d)!}{s!\,(n-d-s)!} = 0 \\ & \text{for } (n-d-s) < 0 \end{cases} \tag{3}$$

and

$$DiceDimFactor = \begin{cases} 1 & \text{if DiceDim} = 0 \\ \sum_{i_{n-(d-1)}=1}^{n-(d-1)} d_{i_{n-(d-1)}} \times \cdots \times & \\ \sum_{i_{n-(d-1)+j}=i_{n-(d-1)+(j-1)}+1}^{n-(d-1)+j} d_{i_{n-(d-1)+j}} \times \cdots \times & \text{if DiceDim} > 0 \\ \sum_{i_n=i_{n-1}+1}^{n} d_{i_n} & \end{cases} \tag{4}$$

The *DiceDimFactor* computes the number of possible dice combinations for a given Dice Dimensionality. Every possible combination can be shown with a different set of free dimensions. Therefore, it is multiplied by the *FreeDimFactor*.

In order to convey a better understanding of how Formula 2 has to be applied, we present it with both parameters *DiceDim* and *FreeDim* set to 2 in Formula 5. The formula applies for all $n \geq 4$.

$$
\begin{aligned}
Views_{2,2} = m \times \Bigg(& \Big((\frac{(n-2)(n-3)}{2} + n - 1) \times \sum_{i=1}^{n-1} d_i \sum_{j=i+1}^{n} d_j \Big) \\
& + \Big((\frac{(n-1)(n-2)}{2} + n) \times \sum_{i=1}^{n} d_i \Big) \\
& + \Big((\frac{n(n-1)}{2} + n + 1) \Big) \Bigg)
\end{aligned} \tag{5}
$$

The maximum number of possible views is massively decreased by the restrictions *DiceDim* and *FreeDim* for high dimensional datasets. In our URL-scheme, these restrictions can be regarded by defining a maximum pattern. The maximum pattern of Formula 5 is “X22”. It means that X *measures2project*, two *dimensions2keep* and two *members2dice* are possible.

We evaluated the formulas numerically and received the same result for Formula 1 and 2 by setting the restrictions to n. We also implemented an algorithm which generates a list of all possible URLs and received an overall number of links as computed by Formula 2. Therefore, we conclude the correctness of the formulas.

3 SEO Evaluation

The previous chapter explained the conceptual model of our approach *SEO4OLAP*. In order to evaluate it, we implemented a concrete system and published two datasets from *Eurostat*. The system is based on Java and deployed on a Google App Engine. In order to evaluate OLAP queries over statistical Linked Data, we used the library OLAP4LD by Kämpgen and Harth [KH14]. The source code of our implementation (for more information, see [Br16]) is published at
`https://github.com/dbreucker/seo4olap`.
For our evaluation, we focused on datasets from Eurostat which were wrapped as Linked Data by Estatwrap. We discovered some challenges that have to be addressed for automated query generation. This is mainly due to the fact that real-world datasets differ from the modelling approaches as intended by the Data Cube Vocabulary (QB) or SDMX. The major issues are presented in the following.

1. Measure-Dimensions: In many datasets by *Eurostat* a dimension is used to specify the indicator of the measure. This means that a measure is modelled as a dimension, a *Measure-Dimension*. We understand that this may be due to a transformation process, e.g. by converting a table into SDMX, and therefore a convenient solution for data publishers. Nevertheless, from a conceptual point of view of a Multidimensional Data Model, this is not intended and causes problems for OLAP-operations. First, instead of a Projection on a measure, a Dice on the *Measure-Dimension* has to be performed. Second, the *Measure-Dimension* cannot be sliced, since an aggregation would cause implausible values. Since QB offers the

possibility to explicitly declare such *Measure-Dimensions*, we conclude that this is a common practice in real-world datasets. Nevertheless, the Linked Data from *Estatwrap* does not make use of this QB-feature.

2. Slice-Members: Some dimensions contain members which are, from a conceptual point of view, aggregations to a higher level. As an example, at *Eurostat* we often find three values for the dimension *Gender*: *Female*, *Male* and *Total*. In consequence, a slice on *Gender* would aggregate all three members and thus lead to wrong values. The member *Total* represents the correct values. Therefore, we define such members as *Slice-Members*. A *Slice-Member* is a member of a dimension, which represents the aggregation of the dimension. We understand that publishers of datasets may have good reasons for this. Nevertheless, this is a challenge for automated OLAP-query generation, since such dimensions cannot be sliced. Instead of a slice, a dice on the *Slice-Member* has to be performed for generating correct values.

In order to evaluate our approach from a SEO perspective, we measured how our generated webpages rank in search engine results in comparison to the original website of *Eurostat*. In the following, we present our evaluation method and our findings. Our assessed evaluation data is published at
`https://github.com/dbreucker/seo4olap-evaluation`.

For the purpose of this evaluation, we acquired the domain http://open-statistics.org. On open-statistics.org, we published two datasets from *Eurostat*. The first is about employment statistics mainly in European countries per year and gender. The second contains information about the gross domestic product in European countries per year. They were first published by us on 22nd of December 2015. The sitemap was submitted to Google on the same day. Since it takes a couple of weeks until new pages rank well, we waited until the 6th of March 2016 for this evaluation. Besides setting two links from other websites, we did not do any SEO off-page optimization techniques in order to strengthen the PageRank or TrustRank of open-statistics.org.

In this evaluation, we measured the search engine rank depending on different keywords for our website open-statistics.org and two benchmarks. The baseline is an online pivot table by *Eurostat*, which allows users to explore the dataset. The second benchmark is a website by *Eurostat* which describes a small subset of facts included in the datasets. It contains a lot of text and therefore many keywords.

Since Google is by far the most used search engine, we only measured the Google rank. The rank assessment was done by the software CuteRank. By using the software, we guarantee that the rank is not influenced by a personal search profile. Ranks higher than 100 are set to 100 for mathematical aggregation.

We defined a set of 84 different search queries, which are grouped by four main keywords. The main keywords were: ”employment”, ”employment rate”, ”gross domestic product” and ”gdp”. So there were two main keywords for each dataset. We considered multiple words as one keyword. Since we wanted to analyse whether the number of keywords per query affects the search engine rank, we further grouped the query-set by the number of included keywords. As an example, the set includes the query ”gdp per capita hungary per year”, which has the main keyword ”gdp” and three other keywords ”per capita”, ”hungary” and ”per year”, thus four keywords in total.

We aggregated the values by building the average value per main keyword depending on the number of keywords. Since queries with a value of 100 strongly influence the average value, we calculated two average values: a normal one including such outliers and a clean one excluding such outliers.

Figure 3 illustrates the Google ranks for the cleaned averages per main keyword depending on the number of keywords. Our own published datasets on open-statistics.org are marked with a dotted line. The corresponding *Eurostat* landing pages as benchmark are illustrated in the same colour with a normal line. The baseline benchmark was never found for any query and is therefore always on top at rank 100.

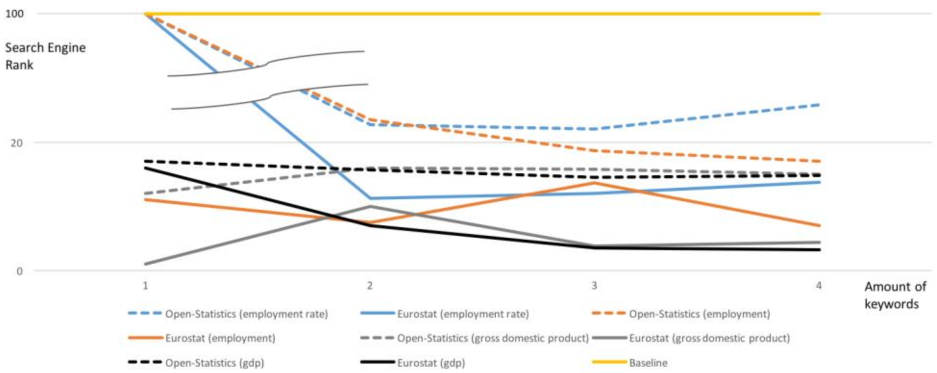

Fig. 3: Cleaned average Google ranks per main keyword for open-statistics.org and benchmarks

Prior to the assessment, we formulated hypotheses, that are presented and discussed in the following:

- Webpages generated by *SEO4OLAP* are indexed by search engines and retrieved for specific queries: The assessment shows that this thesis holds. Even though we created a lot of highly similar webpages, the according landing page for its specified keywords is in most cases retrieved by Google. This shows that the approach to generate websites for a high number of views per dataset is feasible.

- The more specific a query is, the better our search engine results are: Our approach is to generate a lot of views per dataset and thereby produce very specific landing pages for longtail queries, i.e., searches for very specific information that only few websites offer. Since a query consisting of more keywords is more specific and thereby has less competition, we assumed that our pages rank better with more keywords. This thesis cannot be confirmed by the assessed data. As soon as two keywords are involved in a query, the rankings do not improve significantly with further keywords.

- *SEO4OLAP* pages rank better than the benchmark: This thesis holds depending on the benchmark. The baseline, i.e. the pivot table containing the dataset, is never found by Google for our defined queries. In comparison to this, our pages rank significantly better. On the contrary, the manually generated landing pages by *Eurostat* rank very well and on average always better than our pages.

4 Discussion

The idea of our approach *SEO4OLAP* is to generate custom SEO-landing pages for every possible view of a data cube. The evaluation of our implementation shows that, by doing this, *SEO4OLAP* was able to achieve better rankings than the baseline. In the case of *Eurostat*, the baseline was an online pivot table showing the dataset. In consequence, our approach is an improvement for data publishers who simply publish their data without further efforts to manually describe their data on landing pages. This leads to the following benefits:

- Webpages created by *SEO4OLAP* are found by Google for the corresponding keywords. This can be a new source of user traffic for dataset publishers.
- Single facts or views of a dataset are presented in a human readable representation. Thereby, facts of the Semantic Web are made accessible to normal web surfers.
- *SEO4OLAP* allows to reference specific facts in an HTML representation by a URL, whereas before, one could only refer to an entire dataset. As an example, this is an advantage for researcher who want to reference a statistic source.

In the following, some aspects regarding our approach and our evaluation are discussed.

Besides the baseline, we also benchmarked our approach against manually created landing pages by *Eurostat*. Our evaluation shows that on average, we were not able to achieve better rankings than this benchmark. It has to be noted that *Eurostat* has a high authority domain which is trusted by Google. Our rankings would assumably be better, if we had done SEO off-page optimization to gain domain trust and PageRank. Nevertheless, we can only speculate whether we could have beaten the benchmark by applying these techniques. But we can derive a recommendation for data publishers. If the goal of using *SEO4OLAP* is to gain a new source of traffic, we recommend to apply off-page optimization techniques in order to achieve rankings on the first search engine result page.

In contrast to manually generating descriptive landing pages, our approach allows to automatically generate potentially thousands of webpages. From a SEO perspective, this is a longtail strategy, since these pages are optimized for very specific search queries with low competition. As mentioned in our evaluation, we assumed that the more keywords are added to a query, the more specific the query and thus the better the rank. We observed that, once a query includes two keywords, adding further keywords does not improve the rank. An explanation would be that queries consisting of two keywords are already very specific in the domain of statistical facts. Thus, the competition for these keywords is already very low. At this point, our pages compete against pages which are not optimized for this exact set of keywords, but instead have a higher PageRank or TrustRank. Therefore, a further specification is not improving the result rank.

In our evaluation, we only published small datasets with a maximum of three dimensions, in order to test our approach at first with reduced complexity. Therefore, we do not know how search engines react to high dimensional datasets with more than 100.000 pages for one dataset. We leave the SEO-evaluation of high dimensional datasets to future research.

SEO4OLAP converts machine readable data into human readable webpages. Search engines would benefit, if the data on webpages was also provided in a machine readable manner. Therefore, semantic mark-up technologies such as *Schema.org* were developed. In our implementation, we added *Schema.org* mark-up in order to semantically describe the content of our webpages. The main intention of doing this, was to achieve better search engine rankings. To the best of our knowledge, *Schema.org* does not provide means to properly describe statistic facts. It is possible to describe a dataset, the publisher, the publishing date and many more. However, single statistical facts cannot be described. We think our approach and also statistical data itself would benefit, if such functionality would be added to *Schema.org*.

5 Related Work

This is, to the best of our knowledge, the first attempt to generate search engine optimized landing pages from data cubes, published as Linked Data. The related work can be grouped in 1) the visualization and human-friendly interaction with Linked Data and 2) studies about Search Engine Optimization. In the following, we discuss some of those studies.

Since Linked Data has a self-describing data format, it has the benefit of being machine-readable, thus allowing machine interpretation, e.g. by search engines such as the Google Knowledge Graph. On the contrary, even though the amount of published Linked Data is growing on a fast scale, there is still a lot of research to be conducted on how non-experts can interact with this data. One approach is described by Hoefler [Ho13]. He introduces different tools, that allow the analysis and visualization of Linked Data without the knowledge of SPARQL or other semantic technologies. Those tools, further explained by Sabol et al [Sa14] are e.g. the CODE Query Wizard and the CODE Visualization Wizard (Vis Wizard). A similar approach is presented by Salas et al. [Sa12] with CubeViz. The presented studies analyse, transform or visualize RDF Data Cubes for direct user interaction. In contrast, we focus on a presentation optimized for search engines.

The other related field deals with Search Engine Optimization. A lot of research in this area is conducted by major SEO-Agencies, who drive experiments in order to further understand how website rankings can be improved. Since they rely on a competitive advantage, it can be concluded that only a fraction of these experiments are made public. Nevertheless, some academic studies are available. Contrary to our approach, they analyse which techniques influence the search engine ranking; we apply these techniques for our approach. Beel et al. [BGW09] analyse how the ranking of academic papers can be improved. They present some advice for optimization within papers, but do not evaluate whether their approach is successful. Shih et al. [SCC13] describe an empirical evaluation on how basic SEO-techniques correspond to rankings and Malaga [Ma09] evaluates the effect of Web 2.0 techniques.

6 Conclusion & Future Work

We have presented our approach *SEO4OLAP* to generate search engine optimized landing pages from arbitrary data cubes, modelled in the RDF Data Cube Vocabulary. We developed a new OLAP query model which can be represented by a clean URL-scheme and therefore allows to be submitted via HTTP. We illustrated a system architecture which is able to process such OLAP-requests by transforming them into *Logical OLAP Query Plans*, executes them on an OLAP-engine and enhances results with corresponding keywords.

We presented two formulas to calculate the number of possible views depending on the number of members per dimension. The first formula calculates the upper bound of possible views and is exponentially growing with the number of dimensions; the second formula introduces two restrictions in order to decrease the overall number. Both formulas were numerically verified.

To evaluate our approach from a SEO perspective, we implemented a *SEO4OLAP* system in Java and published two datasets from *Eurostat* which resulted in 614 generated landing pages. We evaluated how well our pages are found by search engines in comparison to the dataset source, as well as to manually generated websites from *Eurostat*.

As a conclusion, it can be stated that our approach is feasible in practice and has benefits for data publishers, search engine providers and web users. First, single facts of a dataset are represented as HTML and are thus referenceable and human readable; Second, facts can be found by search engines which results in a new traffic source for data publishers; Third, the system provides an interface for non-experts to statistical Linked Data.

As future work, we plan to publish more data cubes, e.g., all 5,000 datasets from Eurostat; performance optimisations will be necessary to generate and query several GB of RDF data. Also, we plan to investigate the possibility to automatically increase the usefulness of our landing pages by providing additional information from the Semantic Web, e.g., the more commonly used dimension member label “Women” instead of “Female”.

Acknowledgements. This work was supported by the German Ministry of Education and Research (BMBF) within the BigGIS project (Ref. 01IS14012A).

References

[BGW09] Beel, J.; Gipp, B.; Wilde, E.: "Academic Search Engine Optimization (aseo) Optimizing Scholarly Literature for Google Scholar and Co." Journal of scholarly publishing, vol. 41, no. 2, pp. 176–190, 2009.

[Br16] Breucker, D.: "Search Engine optimized presentation of Statistical Linked Data". Master Thesis at Karlsruhe Institute of Technology, 2016.

[Ho13] Hoefler, P.: "Linked data interfaces for non-expert users," in The Semantic Web: Semantics and Big Data. Springer, pp. 702–706, 2013.

[KH14] Kämpgen, B.; Harth, A.: "OLAP4LD - A Framework for Building Analysis Applications over Governmental Statistics" ESWC 2014 Posters & Demo session, Springer, Mai, 2014.

[KOH12] Kämpgen, B.; O'Riain, S.; Harth, A.: "Interacting with statistical linked data via OLAP operations," in The Semantic Web: ESWC 2012 Satellite Events. Springer, pp. 87–101, 2012.

[Ma09] Malaga, R.: "Web 2.0 Techniques for search engine optimization: Two case studies," Review of Business Research, vol. 9, no. 1, pp. 132–139, 2009.

[Mu14] Mutlu, B. et.al.: "Suggesting visualisations for published data," Proceedings of IVAPP, pp. 267–275, 2014.

[Sa12] Salas, P. E. R. et.al.: "Publishing statistical data on the web," in Semantic Computing (ICSC), 2012 IEEE Sixth International Conference on. IEEE, pp. 285–292, 2012.

[Sa14] Sabol, V. et.al.: G. "Discovery and visual analysis of linked data for humans," in The Semantic Web–ISWC 2014. Springer, pp. 309–324, 2014.

[SCC13] Shih, B.-Y.; Chen, C.-Y.; Chen, Z.-S.: "An empirical study of an internet marketing strategy for search engine optimization", Human Factors and Ergonomics in Manufacturing & Service Industries, vol. 23, no. 6, pp. 528–540, 2013.