Representing verifiable statistical index computations as linked data

Jose Emilio Labra Gayo
University of Oviedo
Dept. of Computer Science
C/Calvo Sotelo, S/N
labra@uniovi.es

Hania Farham
The Web Foundation
hania@webfoundation.org

Juan Castro Fernández
WESO Research Group
University of Oviedo
C/Calvo Sotelo, S/N
juan.castro@weso.es

Jose María Álvarez Rodríguez
Dept. Computer Science
Carlos III University
Leganés, Madrid
josemaria.alvarez@uc3m.es

ABSTRACT

In this paper we describe the development of the Web Index linked data portal that represents statistical index data and computations.

The Web Index is a multi-dimensional measure of the World Wide Web's contribution to development and human rights globally. It covers 81 countries and incorporates indicators that assess several areas like universal access; freedom and openness; relevant content; and empowerment.

In order to empower the Web Index transparency, one internal requirement was that every published data could be externally verified. The verification could be that it was just raw data obtained from an external source, in which case, the system must provide a link to the data source or that the value has been internally computed, in which case, the system provides links to those values. The resulting portal contains data that can be tracked to its sources so an external agent can validate the whole index computation process.

We describe the different aspects on the development of the WebIndex data portal, which also offers new linked data visualization tools.

Although in this paper we concentrate on the Web Index development, we consider that this approach can be generalized to other projects which involve the publication of externally verifiable statistical computations.

Categories and Subject Descriptors

H.2.8 [**Database Applications**]: Statistical databases; H.3.5 [**Online Information Services**]: Web-based services

General Terms

Theory

Keywords

Linked data, Statistics, Computations, RDF

1. INTRODUCTION

The creation and use of quantitative indexes is a widely accepted practice that has been applied to numerous domains like economics and Bibliometrics (Impact factor), research and academic performance (H-Index or Shanghai rankings), cloud computing (Global Cloud Index, by CISCO), etc.

We consider that those indexes could benefit from a Linked Data approach where the rankings could be seen, tracked and verified by their users.

We participated in the Web Index project (http://thewebindex.org), which created an index to measure the Web impact in different countries.

The 2012 version offered a data portal[1] whose data was obtained by transforming raw observations and precomputed values from Excel sheets to RDF [2].

In the 2013 version of that data portal, we are working on both validating and computing observations to automatically generate the index from raw data.

We have defined a generic vocabulary of computational index structures which could be applied to compute and validate any other kind of index and can be seen as an specialization of the RDF Data Cube vocabulary [8].

The validation process employs a combination of SPARQL [9] queries and Shape Expressions [3] to check the different integrity constraints and computation steps in a declarative way.

Given that the most important part of a data portal about statistical indexes are the numeric values of each observations we established the internal requirement that any value published should be justified

[1] http://data.webfoundation.org

either declaring from where it had been obtained or linking it to the values of other observations from which it had been computed.

The resulting data portal http://data.webfoundation.org/webindex/2013 contains not only a linked data view about the statistical data but also a machine verifiable justification of the index ranks.

Along the paper we will use Turtle and SPARQL notation and assume that the namespaces have been declared using the most common prefixes found in http://prefix.cc.

2. WEBINDEX COMPUTATION PROCESS

The Web Index is a composite measure that summarizes in a single (average) number the impact and value derived from the Web in various countries. There are serious challenges when attempting to measure and quantify some of the dimensions the Index covers (e.g. the social and political), and suitable proxies were used instead.

As the Web Index covers a large number of countries, some of which have serious data deficiencies or were not covered by the data providers, we needed to impute the missing data. A number of experts in the relevant fields were selected to overcome these challenges and produce a robust and rigorous Index.

Two types of data were used in the construction of the Index: existing data from other data providers (*secondary data*), and new data gathered via a multi-country questionnaire (*primary data*) that was specifically designed by the Web Foundation and its advisers. These primary data will begin to fill in some of the gaps in measurement of the utility and impact of the Web in various countries. Indeed, the data gaps in this field are significant, and we aim to expand those questionnaires in future editions of the Index, both in terms of the questions/indicators gathered and the number of countries covered by the Index.

There are several steps in the process of constructing a composite Index. Some of those involve deciding which statistical method to use in the normalisation and aggregation processes. In arriving at that decision, we took into account several factors, including the purpose of the Index, the number of dimensions we were aggregating, and the ease of disseminating and communicating it, in an understandable, replicable and transparent way. The following 10 steps summarise the computation process of the Index:

1. Take the data for each indicator from the data source for the 81 countries covered by the Index for the 2007-2012 time period (or 2013, in the case of the Web Index expert assessment survey).

2. Impute missing data for every secondary indicator for the sample of 81 countries over the period 2007-2012. Some indicators were not imputed as it was not logical to do so. None of the primary data indicators were imputed. Hence the 2013 Index is very different from the 2007-2012 Indexes that were computed using secondary data only.

   Broadly, the imputation of missing data was done using two methods: country-mean substitution if the missing number is in the middle year (e.g. have 2008 and 2010 but not 2009), or taking arithmetic growth rates on a year-by-year basis. Most missing data for 2011 and 2012 are imputed by applying the arithmetic growth rate for the period, to the 2010 number. For the indicators that did not cover a particular country in any of the years, no imputation was done for that country/indicator.

3. Normalise the full (imputed) dataset using z-scores, making sure that for all indicators, a high value is *good* and a low value is *bad*.

4. Cluster some of the variables (as per the scheme in the tree diagram), taking the average of the clustered indicators post normalisation. For the clustered indicators, this clustered value is the one to be used in the computation of the Index components.

5. Compute the nine component scores using arithmetic means, using the clustered values where relevant.

6. Compute the min-max values for each z-score value of the components, as this is what will be shown in the visualisation tool and other publications containing the component values (generally, it is easier to understand a min-max number in the range of 0 – 100 rather than a standard deviation-based number). The formula for this is: $\frac{x-min}{max-min} \times 100$

7. Compute sub-index scores by calculating the weighted averages of the relevant components for each sub-Index.

8. Compute the min-max values for each z-score value of the sub-Indexes, as this is what will be shown in the visualization tool and other publications containing the sub-index values.

9. Compute overall composite scores by calculating the weighted average of the sub-indexes.

10. Compute the min-max values (on a scale of 0-100) for each z-score value of the overall composite scores, as this is what will be shown in the visualisation tool and other publications containing the composite scores.

The computation process was originally done by human experts using the Excel file although once the process was established, we considered that it could be automated so we could validate the whole process.

3. WEBINDEX WORKFLOW

The WebIndex workflow has been depicted in figure 1.

The Excel file was comprised of 184 Excel sheets and contained a combination of raw, imputed and normalized data created by human experts using Excel.

That external data filtered and converted to RDF by means of an specialized web service called *wiFetcher*.

Although some of the imported values had been pre-computed in Excel by human experts, we collected only the raw values, so we could validate the computation process.

In this way, another application called *wiCompute* took the raw values and computed the index following the computation steps defined by the experts. *wiCompute* carried out the computations generating RDF datasets for the intermediary results and linking the generated values to the values from which they had been computed.

Finally, the RDF data generated was published to a SPARQL endpoint from which we created a specialized visualization tool called *Wesby*.

4. WEBINDEX DATA PORTAL REPRESENTATION

Given the statistical nature of the data, we opted by reusing the RDF Data Cube vocabulary.

Our data model consists of a list of observations which can be raw observations obtained from an external source or computed observations derived from other observations. An example observation can be:

```
obs:obsM23 a qb:Observation ;
 cex:computation [ a cex:Z-Score ;
     cex:observation obs:obsA23 ; cex:slice
         slice:sliceA09 ; ] ;
 cex:value 0.56 ;
 cex:md5-checksum "2917835203..." ;
 cex:indicator indicator:A ;
 cex:concept country:ESP ;
 qb:dataSet dataset:A-Normalized ;
 # ... other declarations omitted for
     brevity
```

Where we declare that obs:obsM23 is an observation whose value is 0.56 that has been obtained as the Z-Score of the observation obs:A23 using the slice slice:sliceA09. The observations refers to indicator indicator:A, to the concept country:ESP and to the dataset dataset:A-Normalized.

For each observation, we also add a value for cex:md5-checksum which is obtained as a combination of the different values of the observation and allows a user to verify the values asserted to that observation.

5. COMPUTEX VOCABULARY

The *Computex* vocabulary is available at http://purl.org/weso/computex. It defines terms related to the computation of statistical index data and is compatible with RDF Data Cube vocabulary. Some terms defined in the vocabulary are:

- **cex:Concept** represents the entities that we are indexing. In the case of the Web Index project, the concepts are the different countries. In other applications it could be Universities, journals, services, etc.
- **cex:Indicator**. A dimension whose values add information to the Index. Indicators can be simple dimensions, for example: the mobile phone suscriptions per 100 population, or can be composed from other indicators.
- **qb:Observation**. This is the same term as in the RDF Data Cube vocabulary. It contains values for the properties: cex:value, cex:indicator and cex:concept, etc. The value of a qb:Observation can be a Raw value obtained from an external source or a computed value obtained from other observations.
- **cex:Computation**. We have declared the main computation types that we needed for the WebIndex project, which have been summarized in Table 5. That list of computation types is non-exhaustive and can be further extended in the future.
- **cex:WeightSchema** a weight schema for a list of indicators. It consists of a weight associated for each indicator which can be used to compute an aggregated observation.

6. VALIDATION APPROACH

The validation approach employed in the 2012 WebIndex project was based on resource templates similar to the OSLC resource shapes[2] and the MD5 checksum field. Apart from that, we did not verify that the precomputed values imported from the Excel sheets really match the value that could be obtained by following the declared computation process.

We propose a new validation approach that goes a step forward. The goal is not only to check that a resource contains a given set of fields and values, but also that those values really match the values that can be obtained by following the declared computations.

The proposed approach has been inspired by the integrity constraint specification proposed by the RDF Data Cube vocabulary which employs a set of SPARQL ASK queries to check the integrity of RDF Data Cube data. Although ASK queries provide a good means to check integrity, in practice their boolean nature does not offer too much help when a dataset does not accomplish with the data model.

We decided to use CONSTRUCT queries which, in case of error, contain an error message and a list of error parameters that can help to spot the problematic data.

We transformed the ASK queries defined in the RDF Data Cube specification to CONSTRUCT queries. For example, the query to validate the RDF Data Cube integrity constraint 4 (IC-4) is:

```
CONSTRUCT {
 [ a cex:Error ; cex:errorParam [cex:name
     "dim"; cex:value ?dim ] ;
   cex:msg "Every Dimension must have a
       declared range" . ]
} WHERE { ?dim a qb:DimensionProperty .
 FILTER NOT EXISTS { ?dim rdfs:range [] }
}
```

In order to make our error messages compatible with EARL [1], we have defined cex:Error as a subclass of earl:TestResult and declared it to have the value earl:failed for the property earl:outcome.

We have also created our own set of SPARQL CONSTRUCT queries to validate the *Computex* vocabulary terms, specially the computation of index data. For example, the following query validates that every observation has at most one value.

```
CONSTRUCT {
 [ a cex:Error ; cex:errorParam # ...
     omitted
    cex:msg "Observation has two different
        values" . ]
} WHERE { ?obs a qb:Observation .
 ?obs cex:value ?value1 . ?obs cex:value
     ?value2 .
 FILTER ( ?value1 != ?value2 )
}
```

[2] http://www.w3.org/2012/12/rdf-val/SOTA

Computation	Description	Properties
Raw	No computation. Raw value obtained from external source.	
Mean	Mean of a set of observations	`cex:observation` `cex:slice`
Increment	Increment an observation by a given amount	`cex:observation` `cex:amount`
Copy	A copy of another observation	`cex:observation`
Z-score	A normalization of an observation using the values from a Slice.	`cex:observation` `cex:slice`
Ranking	Position in the ranking of a slice of observations.	`cex:observation` `cex:slice`
AverageGrowth	Expected average growth of N observations	`cex:observations`

WeightedMean	Weighted mean of an observation	`cex:observation`
		`cex:slice`
		`cex:weightSchema`

Table 1: Some types of statistical computations

Using this approach, it is possible to define more expressive validations. For example, we are able to validate that an observation has been obtained as the mean of other observations.

```
CONSTRUCT {
 [ a cex:Error ; cex:errorParam # ...omitted
   cex:msg "Mean value does not match" ] .
} WHERE {
   ?obs a qb:Observation ;
       cex:computation ?comp ;
       cex:value ?val .
  ?comp a cex:Mean .
  { SELECT (AVG(?value) as ?mean) ?comp
      WHERE {
     ?comp cex:observation ?obs1 .
        ?obs1 cex:value ?value ;
  } GROUP BY ?comp }
 FILTER( abs(?mean - ?val) > 0.0001)
}
```

We have implemented an online validation tool called *Computex*[3] which takes as input an RDF graph and checks if it follows the integrity constraints defined by Computex. The validation tool can also check if the RDF graph follows the RDF Data Cube integrity constraints and it can also do the index computation for RDF Graphs. Although this declarative approach was very elegant, computing the webindex using only SPARQL queries was not practical (it took around 15 minutes for a small subset), so the computation process was finally done by a specialized program implemented in Scala [4].

7. EXPRESSIVENESS LIMITS OF SPARQL QUERIES

Validating statistical computations using SPARQL queries offered a good exercise to check SPARQL expressivity. Although we were able to express most of the computation types, some of them had to employ functions that are not part of SPARQL 1.1 or had to be defined in a limited way. In this section we review some of the challenges that we found.

- The Z-score of a value x_i is defined as $\frac{x-\bar{x}}{\sigma}$ where $\bar{x}$ is the mean and $\sigma = \sqrt{\frac{\sum_{i=1}^{N}(\bar{x}-x_i)^2}{N-1}}$ is the standard deviation. To validate that computation using SPARQL queries, it is necessary to employ the `sqrt` function. This function is not available in SPARQL 1.1 although some implementations like Jena ARQ[5] provide it.

- In order to validate the ranking of an observation (in which position it appears in a list of observations), we have found two approaches. One is to check all the observations that are below the value of that observation. This approach requires checking the value of each observation against all the other values. The other approach is to use a subquery that groups all the observations ordered by their value using the `GROUP_CONCAT`. However, SPARQL does not offer a function to calculate the position of a substring in a string[6], so we divided the length of the substring before the concept's name by the length of the concept's name. This approach is more efficient but only works when all the names have the same length.

- Given a list of values $x_1, x_2 \ldots x_n$ the expected value x_{n+1} can be extrapolated using the forward average growth formula: $x_n \times \frac{\frac{x_n}{x_{n-1}}+\ldots+\frac{x_2}{x_1}}{n-1}$. Accessing RDF collections in SPARQL 1.1 requires property paths and offers limited expressivity. In this particular case the query can be expressed as[7]:

```
CONSTRUCT {
  # ... omitted for brevity
} WHERE {
  ?obs cex:computation [a
       cex:AverageGrowth;
       cex:observations ?ls] ;
  cex:value ?val .
  ?ls rdf:first [cex:value ?v1 ] .
  { SELECT ( SUM(?v_n / ?v_n1)/COUNT(*)
       as ?meanGrowth) WHERE {
```

[3] http://computex.herokuapp.com/

[4] Source code is available here: https://github.com/weso/wiCompute

[5] http://jena.apache.org/documentation/query/library-function.html

[6] This function is called `strpos` in PHP or `indexOf` in Java

[7] This query was suggested by Joshua Taylor.

```
    ?ls rdf:rest* [ rdf:first [
        cex:value ?v_n ] ;
                rdf:rest [ rdf:first [
                    cex:value ?v_n1 ]]]
                    .
 }}
 FILTER (abs(?meanGrowth * ?v1 - ?val)
    > 0.001) }
```

8. DOCUMENTATION USING SHAPE EXPRESSIONS

In order to document the resulting data portal we created a set of templates using Shape Expressions [8]. We consider that this approach offer a good balance between human readability and machine processable specification.

As an example, the Shape Expression declaration of observations is:

```
<Observation> {
  rdf:type ( qb:Observation )
, rdfs:label ( @en ) ?
, cex:md5-checksum xsd:string ?
, cex:computation @<Computation>
, dcterms:issued xsd:date ?
, dcterms:publisher (:WebFoundation)
, qb:dataSet @<DataSet>
, wfonto:ref-area @<Area>
, cex:indicator @<Indicator>
, wfonto:ref-year @<Year>
, cex:value xsd:float
}
```

In fact, we are currently developing an online RDF validation tool called *RDFShape*[9] that can validate if an IRI dereferenced from a data portal has a given shape.

9. VISUALIZING THE DATA PORTAL USING WESBY

Although the main user for a data portal can be considered to be a machine, in practice, data portals are also browsed by human beings using conventional browsers.

Following the linked data principles, we considered that it was necessary to offer not only RDF views but also HTML representations of the different data.

We developed a visualization tool called Wesby [10] which takes as input an SPARQL endpoint and offers a linked data browsing experience. Wesby was inspired by Pubby [7] and was developed in Scala using the Play! Framework. Wesby handles combines the visualization with a set of templates to offer specialized views for different types of resources. For example, figure 2 contains the WebIndex visualization of the country Spain.

When there is no template for a given type of node, Wesby shows a table of properties and values similar to Pubby. For example, figure 3

10. RELATED WORK

There is a growing interest in developing solutions to improve the quality of linked data [10, 14, 11]. We consider that is is very important to publish linked data that is not only of high quality, but also that can automatically be validated. Validating RDF has also attracted a number of approaches. Most of them were presented at the W3c Workshop on RDF Validation [16] and can be summarized as:

Inference based approaches which try to adapt RDF Schema or OWL. However, the use of Open World and Non-unique name assumption limits the validation possibilities. [5, 18, 15] propose the use of OWL expressions with a Closed World Assumption to express integrity constraints.

SPARQL queries can express the validation constraints. SPARQL has much more expressiveness than Shape Expressions and was proposed in [12] to validate the WebIndex before it was generated. After the WebIndex has been launched we noticed that validating the whole WebIndex data portal using the SPARQL approach was not efficient so we have been involved in the development of Shape Expressions.

Grammar based approaches define a domain specific language to declare the validation rules. OSLC Resource Shapes [17] have been proposed as a high level and declarative description of the expected contents of an RDF graph expressing constraints on RDF terms. Dublic Core Application Profiles [6] also define a set of validation constraints using Description Templates. Finally, Shape Expressions have been proposed as another technology inspired by RelaxNG [19] and based on regular bag expresions [3].

Representing statistical linked data has also seen an increasing interest. SDMX [11] is the primary format of the main statistical data organizations. In [4] describes a the transformation of SDMX-ML data to RDF/XML. The RDF Data Cube vocabulary [8] has been accepted as a W3c Recommendation technology to publish multidimensional statistical data and to link it with other concepts and data. We have opted to follow the RDF Data Cube vocabulary and in fact, we consider that Computex can be seen as a further specialization of RDF Data Cube to represent statistical index computations.

Another piece of related work is how to represent mathematical expressions in RDF. Lange [13] gives an overview of the different approaches. OpenMath was proposed as an extensible standard that can represent the semantic meaning of mathematical objects. Wenzel and Reinhardt [20] propose an approach to integrate OpenMath with RDF data for the representation of mathematical relationships and the integration of mathematical computations into reasoning systems.

11. CONCLUSIONS

In this paper, we described how we were able to represent statistical index computations as linked data which include information to track the origin of any published observation.

Although the number of triples were around 3 million, we consider that the data portal is of medium size, so we were able to play with different validation possibilities. In fact, although we initially wanted to do the whole computation process using SPARQL

[8] http://weso.github.io/wiDoc/

[9] http://rdfshape.weso.es

[10] http://wesby.weso.es

[11] http://sdmx.org/

CONSTRUCT queries, we found that it took longer than expected (around 15 minutes for a subset of the whole dataset) so we opted to develop an independent Scala program in Scala that did all the computation in a few seconds.

After participating in the W3c RDF Validation workshop we were attracted by the Shape Expressions formalism. We developed the documentation of the data portal using Shape Expressions and we have developed an RDF validation tool using it. We consider that some structural parts of the data portal can be better expressed in Shape Expressions, while the computational parts could be expressed using Shape Expression semantic actions, all of which could be finally translated to SPARQL.

Using SPARQL queries to validate and compute index data seems a promising use case for linked data applications. Although we have successfully employed this approach to validate most of the statistical computations we needed for the WebIndex project, we have found some limitations in current SPARQL 1.1 expressivity with regards to built-in functions on maths, strings and RDF Collections.

Our future work is to automate the declarative computation of index data from the raw observations and to check the performance using the Web Index data. We are also studying the feasibility of this approach for online calculation of index scores and rankings.

12. ACKNOWLEDGEMENTS

We would like to thank Jules Clements, Karin Alexander, César Luis Alvargonzález and Ignacio Fuertes Bernardo for their collaboration in the development of the WebIndex project.

References

[1] S. Abou-Zahra. Evaluation and Report Language EARL 1.0 schema. http://www.w3.org/TR/EARL10-Schema/, 2011. W3C Working Draft.

[2] J. M. Alvarez Rodríguez, J. Clement, J. E. Labra Gayo, H. Farhan, and P. Ordoñez. *Cases on Open-Linked Data and Semantic Web Applications*, chapter Publishing Statistical Data following the Linked Open Data Principles: The Web Index Project., pages 199–226. IGI Global, 2013. doi:10.4018/978-1-4666-2827-4.ch011.

[3] I. Boneva, J. E. Labra, S. Hym, E. G. Prud'hommeau, H. Solbrig, and S. Staworko. Validating RDF with Shape Expressions. *ArXiv e-prints*, Apr. 2014.

[4] S. Capadisli, S. Auer, and A.-C. Ngonga Ngomo. Linked sdmx data. *Semantic Web Journal*, pages 1–8, 2013.

[5] K. Clark and E. Sirin. On RDF validation, stardog ICV, and assorted remarks. In *RDF Validation Workshop. Practical Assurances for Quality RDF Data*, Cambridge, Ma, Boston, September 2013. W3c, http://www.w3.org/2012/12/rdf-val.

[6] K. Coyle and T. Baker. Dublin core application profiles. separating validation from semantics. In *RDF Validation Workshop. Practical Assurances for Quality RDF Data*, Cambridge, Ma, Boston, September 2013. W3c, http://www.w3.org/2012/12/rdf-val.

[7] R. Cyganiak and C. Bizer. Pubby: A linked data frontend for sparql endpoints. http://www4.wiwiss.fu-berlin.de/pubby/.

[8] R. Cyganiak and D. Reynolds. The RDF Data Cube Vocabulary. http://www.w3.org/TR/vocab-data-cube/, 2013. W3c Candidate Recommendation.

[9] S. Harris and A. Seaborne. SPARQL 1.1 Query Language. http://www.w3.org/TR/sparql11-query/, 2013.

[10] A. Hogan, A. Harth, A. Passant, S. Decker, and A. Polleres. Weaving the pedantic web. In *Linked Data on the Web Workshop (LDOW2010) at WWW'2010*, volume 628, pages 30–34. CEUR Workshop Proceedings, 2010.

[11] D. Kontokostas, P. Westphal, S. Auer, S. Hellmann, J. Lehmann, R. Cornelissen, and A. Zaveri. Test-driven evaluation of linked data quality. In *Proceedings of the 23rd International Conference on World Wide Web*, WWW '14, pages 747–758, Republic and Canton of Geneva, Switzerland, 2014. International World Wide Web Conferences Steering Committee.

[12] J. E. Labra and J. M. Alvarez Rodríguez. Validating statistical index data represented in RDF using SPARQL queries. In *RDF Validation Workshop. Practical Assurances for Quality RDF Data*, Cambridge, Ma, Boston, September 2013. W3c, http://www.w3.org/2012/12/rdf-val.

[13] C. Lange. Ontologies and languages for representing mathematical knowledge on the semantic web. *Semantic Web*, 4(2):119–158, 2013.

[14] P. N. Mendes, H. Mühleisen, and C. Bizer. Sieve: Linked data quality assessment and fusion. In *Proceedings of the 2012 Joint EDBT/ICDT Workshops*, EDBT-ICDT '12, pages 116–123, New York, NY, USA, 2012. ACM.

[15] B. Motik, I. Horrocks, and U. Sattler. Adding Integrity Constraints to OWL. In C. Golbreich, A. Kalyanpur, and B. Parsia, editors, *OWL: Experiences and Directions 2007 (OWLED 2007)*, Innsbruck, Austria, June 6–7 2007.

[16] RDF Working Group W3c. W3c validation workshop. practical assurances for quality rdf data, September 2013.

[17] A. G. Ryman, A. L. Hors, and S. Speicher. OSLC resource shape: A language for defining constraints on linked data. In C. Bizer, T. Heath, T. Berners-Lee, M. Hausenblas, and S. Auer, editors, *Linked data on the Web*, volume 996 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2013.

[18] J. Tao, E. Sirin, J. Bao, and D. L. McGuinness. Integrity constraints in OWL. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence (AAAI-10)*. AAAI, 2010.

[19] E. van der Vlist. *Relax NG: A Simpler Schema Language for XML*. O'Reilly, Beijing, 2004.

[20] K. Wenzel and H. Reinhardt. Mathematical computations for linked data applications with openmath. In *Conferences on Intelligent Computer Mathematics, CICM 2012*, 2012.

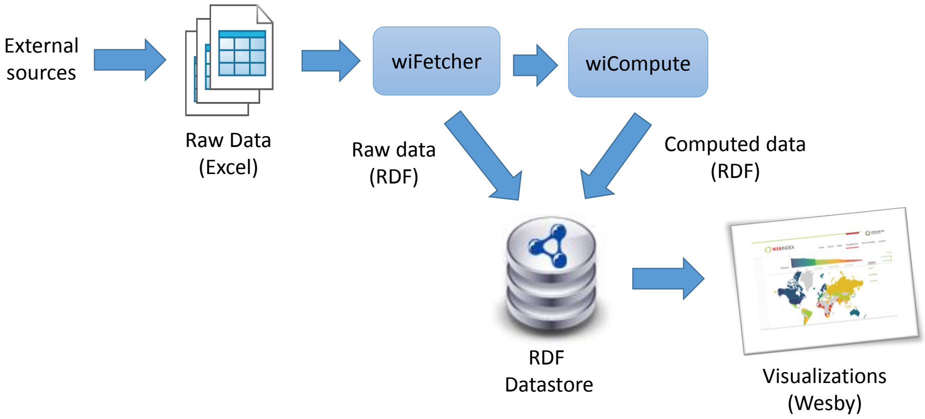
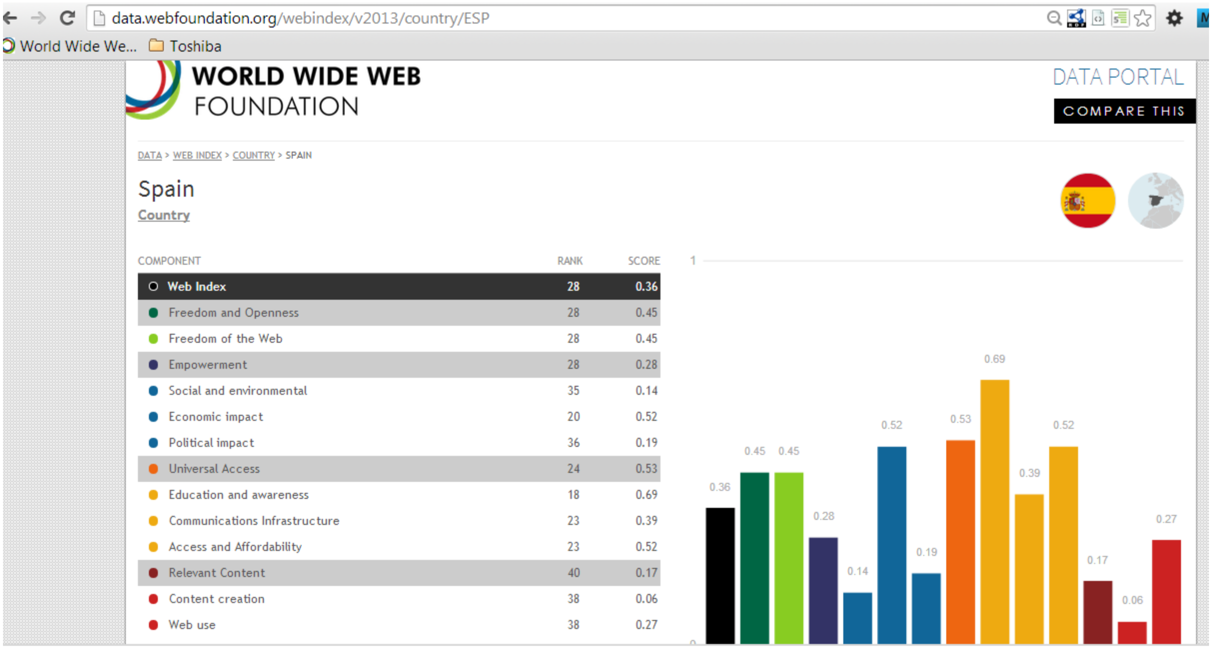


Figure 1: Web Index data portal WorkFlow



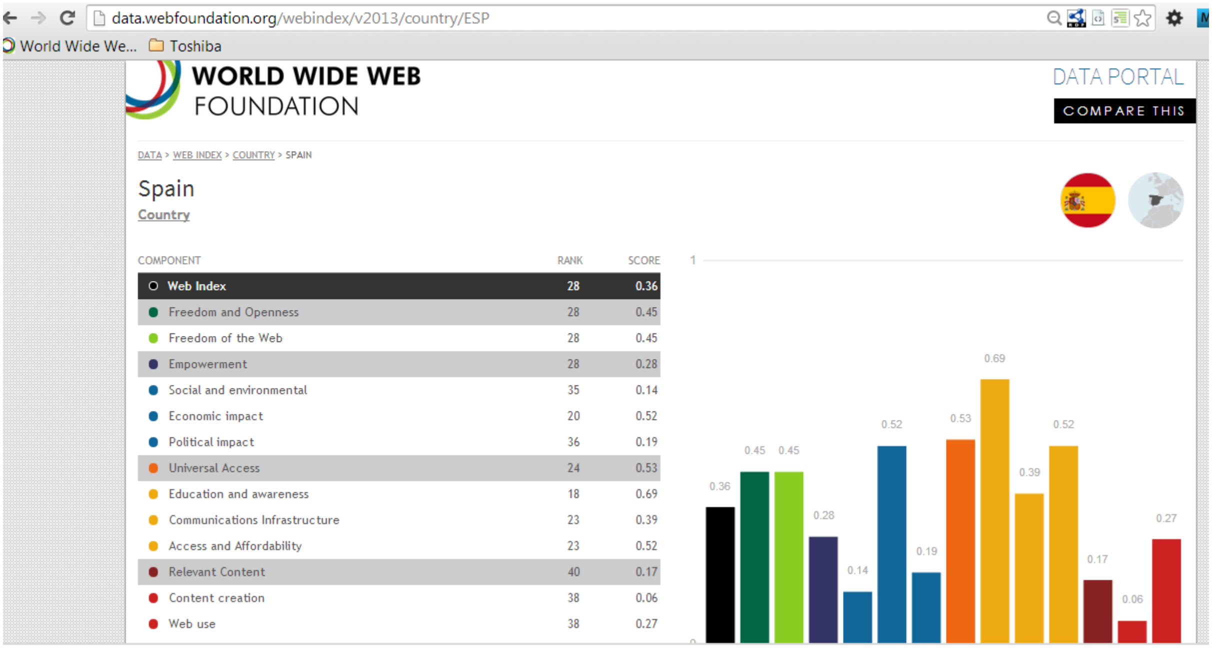
DATA > WEB INDEX > COUNTRY > SPAIN

Spain

Country

COMPONENT	RANK	SCORE
○ Web Index	28	0.36
● Freedom and Openness	28	0.45
● Freedom of the Web	28	0.45
● Empowerment	28	0.28
● Social and environmental	35	0.14
● Economic impact	20	0.52
● Political impact	36	0.19
● Universal Access	24	0.53
● Education and awareness	18	0.69
● Communications Infrastructure	23	0.39
● Access and Affordability	23	0.52
● Relevant Content	40	0.17
● Content creation	38	0.06
● Web use	38	0.27

Figure 2: Web Index visualization of country Spain



data.webfoundation.org/webindex/v2013/country/ESP
World Wide We...
Toshiba
WORLD WIDE WEB FOUNDATION
DATA PORTAL
COMPARE THIS
DATA > WEB INDEX > COUNTRY > SPAIN

Spain

Country

COMPONENT	RANK	SCORE
○ Web Index	28	0.36
● Freedom and Openness	28	0.45
● Freedom of the Web	28	0.45
● Empowerment	28	0.28
● Social and environmental	35	0.14
● Economic impact	20	0.52
● Political impact	36	0.19
● Universal Access	24	0.53
● Education and awareness	18	0.69
● Communications Infrastructure	23	0.39
● Access and Affordability	23	0.52
● Relevant Content	40	0.17
● Content creation	38	0.06
● Web use	38	0.27

Figure 3: Web Index visualization of a weight map