Legislative data portals and linked data quality

Jose Emilio Labra Gayo
WESO
WEB Semantics Oviedo
University of Oviedo, Spain
About me...

In 2004, founded WESO (Web Semantics Oviedo) research group

Goal: Practical applications of semantic technologies

Several domains: e-Government, e-Health,...

2 books:

"Web semántica" (in Spanish), 2012
"Validating RDF data", 2017

...and software:

SHaclEX (Scala library, implements ShEx & SHACL)
RDFShape (RDF playground)
WikiShape (Wikidata playground)

HTML version: http://book.validatingrdf.com
Examples: https://github.com/labra/validatingRDFBookExamples
About this talk…

It will be divided in 2 parts

1st part

Legislative linked data portals
Chilean National library of Congress
Presented at ESWC'19*

2nd part

Linked data quality
RDF description and validation
ShEx & SHACL overview
Legislative linked data portals

Processing the History of the Law at Chile

Francisco Cifuentes Silva
Library of Congress, Chile
PhD Student
WESO research group

Jose Emilio Labra Gayo
WESO research group
University of Oviedo, Spain

More information:
Chilean Library of Congress

In Spanish: BCN (Biblioteca del Congreso Nacional de Chile)

Political powers

- Legislative
- Judiciary
- Executive

Independent body inside the Legislative power
Advices the parliament and gives services to citizens

http://www.bcn.cl
2 projects at library of congress (BCN)

History of the Law
Parliamentary work
History of the Law (LeyChile)

Collect all documents generated during a law legislative process

Phases:

An initiative sees life as a draft bill
Subject to debates
Validity time (it is published)
Modifications, additions,...
Derogation

Goal:
Capture the spirit of the law
Traceability

https://www.bcn.cl/historiadelaley
Parliamentary work

Collect all legislative activity by each Member of Parliament
Retrieve all interventions made
- Parliamentary motion
- Session journal
- Commission report
Ordered and categorised

https://www.bcn.cl/laborparlamentaria/
Both projects adopted semantic technologies

Some initial reasons:

- Semantic technologies considered one pillar of strategic plan (in 2014)
- Innovative action to generate new products
- Improve interoperability mechanisms
- Sem. Web aligned well with open & public data
Which semantic technologies?

Text mining and content enrichment
- Entity extraction
- Topic identification
- Automatic markup
- Classification

Machine readable info
- XML & URIs
- RDF
- Ontologies
- Linked Open Data
Workflow pipelines

3 main steps

- Automatic XML Marker
- RDF & Linked data generation
- Content delivery
Workflow overview

Legislative documents
- Some docs in paper (requires OCR)
- Text documents

National library

Automatic XML marker

XML editor & tools

Linked Open Data

SVN repository
Akoma-Ntoso

Publishing
(RDF extraction
From Akoma-Ntoso)

Services layer

Linked Open Data

Query DB

Content portals
Automatic XML marker

Source: Text

Target: XML following Akoma-Ntoso
Automatic XML marker

4 phases
1. Named Entity Recognizer

Detection of entities & types of entities
Web service implementing the Stanford NER with a CRF classifier
Evaluation in production: detects 97% entities

<table>
<thead>
<tr>
<th>Type</th>
<th>Some examples</th>
<th># of entities</th>
</tr>
</thead>
<tbody>
<tr>
<td>Person</td>
<td>Salvador Allende, Sebastián Piñera</td>
<td>5.139</td>
</tr>
<tr>
<td>Organization</td>
<td>Ministerio de Salud, SERNATUR</td>
<td>2.848</td>
</tr>
<tr>
<td>Location</td>
<td>Valparaíso, Santiago de Chile</td>
<td>1.251</td>
</tr>
<tr>
<td>Document</td>
<td>PROYECTOS DE LEY violencia que ha golpeado a las familias El</td>
<td></td>
</tr>
<tr>
<td>Role</td>
<td>&lt;body bestClass=&quot;ROL&quot; knownFraction=&quot;1&quot;&gt;Secretario&lt;/body&gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&lt;class value=&quot;ROL&quot; probability=&quot;0.2201459321&quot;/&gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>, señor</td>
<td></td>
</tr>
<tr>
<td>Events</td>
<td>&lt;body bestClass=&quot;PER&quot; knownFraction=&quot;1&quot;&gt;Raúl Guerrero Guerrero&lt;/body&gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&lt;class value=&quot;PER&quot; probability=&quot;0.83023132&quot;/&gt;</td>
<td></td>
</tr>
<tr>
<td>Law</td>
<td>y el rseecretario,</td>
<td></td>
</tr>
<tr>
<td>Dates</td>
<td>&lt;body bestClass=&quot;PER&quot; knownFraction=&quot;0.5&quot;&gt;don Fernando Parga Santelices&lt;/body&gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&lt;class value=&quot;PER&quot; probability=&quot;0.9555160392&quot;/&gt;</td>
<td></td>
</tr>
</tbody>
</table>
2. Mediator

Entity linking and disambiguation
- Text similarity algorithms
- Based on Apache Lucene
- In-house development
  - Use of context information to narrow list of candidates
  - Custom filters and association heuristics
  - Specialized web services
3. Structural marker

Detect structures in the text
Titles, subtitles, paragraphs, sections,...
Special structure for debates: participation
Regular expressions + custom rules
4. XML converter to Akom-Ntoso

Programmatic approach
Internal XML representation similar to DOM
Each node converted to text in AKN-XML
Quality assurance by human analysts
They review the generated XML documents
2 editors:
   Ad-hoc XML editor
   Commercial editor: LegisPro (Xcential)
Linked data generation

The pilot project (2011) carefully defined a stable URI model
URIs have been maintained since then
    URIs = IDs in the whole system
URIs are dereferentiable
Content negotiation
Custom linked data browser
Documentation (in Spanish)

http://datos.bcn.cl/es/documentacion
AKN2RDF

RDF extraction from Akoma-Ntoso XML

- Custom-made converter (XSL discarded for perceived complexity)
- Each XML tag implemented in one Class
- Extracted data saved into multiple databases (Relational and RDF)
Linked data generation

**Source:** AKN XML documents

http://datos.bcn.cl/recurso/cl/documento/579095.xml

http://datos.bcn.cl/recurso/cl/documento/579095

**Target:** RDF data

http://datos.bcn.cl/recurso/cl/documento/579095/

http://datos.bcn.cl/recurso/cl/documento/579095

Linked data browser (WESO-DESH)
SPARQL endpoint

RDF triples are published as a public SPARQL endpoint

```
PREFIX bcnnoms: <http://datos.bcn.cl/ontologies/bcn-norms#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX dc: <http://purl.org/dc/elements/1.1/>

SELECT DISTINCT ?name (count(?norm) as ?number)
WHERE {
  ?norm bcnnoms:createdBy ?organization .
  ?organization bcnnoms:hasName ?name .
  ?organization rdf:type bcnnoms:GovernmentalOrganization .
  FILTER regex(?name, "^"MUNICIPALIDAD")
}
ORDER BY DESC (?number)
```
Content delivery

Web portals using Open Source Technologies
  CMS (Typo3)
  Python/Java
  Varnish
  Apache Lucene

REST Web service layers which connect to RDF triplestore and DB

Data exports to PDF, Doc and XML formats

URIs of parliamentary profiles = URIs in triplestore
History of the Law portal

https://www.bcn.cl/historiadelaley

Different versions of a law
Each article has a link
Links to Members of Parliament
History of the Law portal

https://www.bcn.cl/historiadelaley
Parliamentary Work

https://www.bcn.cl/laborparlamentaria
Some experimental visualizations

Relationships between laws
Historical Parliament
Parliamentary genealogy (family relationships)
Regions mentioned in laws (legislative hackathon)
Links between laws
Historical parliament

http://datos.bcn.cl/visualizaciones/genealogia-parlamentaria/
Parliamentary genealogy

http://datos.bcn.cl/visualizaciones/genealogia-parlamentaria/consulta.jsp
Regions mentioned by law

Result of a legislative hackathon

In 2010 there was an Earthquake in BioBio region

24,368 documents (nov. 2018)
Number of RDF triples: 28 millions
According to Google analytics
  Average browsing time: 2min 26s
  Visits received 331,481 (nov. 2016-2017) ⇒ 476,241 (nov. 2017-2018)

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Total docs.</th>
<th>Min.</th>
<th>1st Qu.</th>
<th>Median</th>
<th>Mean</th>
<th>3rd Qu.</th>
<th>Max.</th>
<th>Stdev.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Debate Senate</td>
<td>3.614</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Debate Chamber of Deputies</td>
<td>4.298</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bills</td>
<td>2.514</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Others types</td>
<td>13.942</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>24.368</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
...and some findings...

**Question:** why are there some valleys?

**RDF triples generated by year**

**Dictatorship time**

**Session attendance by year**
Lessons learnt

RDF granularity & inference trade-off

- RDF statements + inference (high running times...queries that didn't terminate)
- A priori inferred triples added to triple store (high response times for large docs)
- Small subset of RDF triples (structural parts of docs and metadata)

Performance problems in XML editor browsing long docs (>1000 pages)

Low SPARQL endpoint usage by external apps

- If we could start again, I would recommend ShEx

Personal note: These kind of projects led to my interest in ShEx
Conclusions & future projects

Well designed URIs can act as a perfect glue for interoperability

Automatic workflow pipelines help long-term survival of LD-based projects

  SPARQL endpoint since 2011

Future projects on top of existing ones

  National Budget as Linked data

  Diana Project: Members of Parliament linked to social network analysis

  New portal: User customization & recommender systems
2nd part
Linked data quality and Shapes
RDF, the good parts...

RDF as an integration language
RDF as a *lingua franca* for semantic web and linked data
RDF flexibility & integration
  
  - Data can be adapted to multiple environments
  - Open and reusable data by default
RDF for knowledge representation
RDF data stores & SPARQL
RDF, the other parts

Consuming & producing RDF

- Multiple syntaxes: Turtle, RDF/XML, JSON-LD, ...
- Embedding RDF in HTML
- Describing and validating RDF content
Why describe & validate RDF?

For producers
- Developers can understand the contents they are going to produce
- Ensure they produce the expected structure
- Advertise and document the structure
- Generate interfaces

For consumers
- Understand the contents
- Verify the structure before processing it
- Query generation & optimization
Similar technologies

<table>
<thead>
<tr>
<th>Technology</th>
<th>Schema</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relational Databases</td>
<td>DDL</td>
</tr>
<tr>
<td>XML</td>
<td>DTD, XML Schema, RelaxNG, Schematron</td>
</tr>
<tr>
<td>Json</td>
<td>Json Schema</td>
</tr>
<tr>
<td>RDF</td>
<td>?</td>
</tr>
</tbody>
</table>

Fill that gap
ShEx and SHACL

2013 RDF Validation Workshop
Conclusions of the workshop:

*There is a need of a higher level, concise language for RDF Validation*

ShEx initially proposed (v 1.0)

2014 W3c Data Shapes WG chartered

2017 SHACL accepted as W3C recommendation

2017 ShEx 2.0 released as Community group draft

2019 ShEx adopted by Wikidata
Short intro to ShEx

ShEx (Shape Expressions Language)
Concise and human-readable language for RDF validation & description
Syntax similar to SPARQL, Turtle
Semantics inspired by regular expressions & RelaxNG
2 syntaxes: Compact and RDF/JSON-LD
Official info: http://shex.io
Nodes conforming to `<User>` shape must:

- Be IRIs
- Have exactly one `schema:name` with a value of type `xsd:string`
- Have zero or more `schema:knows` whose values conform to `<User>`
RDF Validation using ShEx

Schema

```xml
<User> IRI {
  schema:name xsd:string ;
  schema:knows @<User> *
}
```

Shape map

- :alice@<User> ✓
- :bob @<User> ✓
- :carol@<User> ✗
- :dave@<User> ✗
- :emily@<User> ✗
- :frank@<User> ✗
- :grace@<User> ✓

Try it (RDFShape): [https://goo.gl/97bYdv](https://goo.gl/97bYdv)
Try it (ShExDemo): [https://goo.gl/Y8hBsW](https://goo.gl/Y8hBsW)
Validation process

**Input:** RDF data, ShEx schema, Shape map

**Output:** Result shape map

```shex
:User {
schema:name xsd:string ;
schema:knows @:User *
}
```

**Shape map**

`:alice@:User, :bob@:User, :carol@:User`

**RDF data**

`:alice schema:name "Alice" ;
schema:knows :alice .

`:bob schema:knows :alice ;
schema:name "Robert".

`:carol schema:name "Carol", "Carole" .

**Result shape map**

`:alice@:User, :bob@:User, :carol@!:User`
Example with more ShEx features

```shex
:AdultPerson EXTRA rdf:type {
  rdf:type [ schema:Person ] ;
  :name xsd:string ;
  :age MinInclusive 18 ;
  :gender [:Male :Female] OR xsd:string ;
  :address @:Address ? ;
  :worksFor @:Company +
}

:Address CLOSED {
  :addressLine xsd:string {1,3}
  :postalCode /[0-9]{5}/
  :state @:State
  :city xsd:string
}

:Company {
  :name xsd:string
  :state @:State
  :employee @:AdultPerson *
}

:State /[A-Z]{2}/

:alice rdf:type :Student, schema:Person ;
  :name "Alice" ;
  :age 20 ;
  :gender :Male ;
  :address [ :addressLine "Bancroft Way" ;
             :city "Berkeley" ;
             :postalCode "55123" ;
             :state "CA" ] ;
  :worksFor [ :name "Company" ;
              :state "CA" ;
              :employee :alice ].
```

Try it: [https://tinyurl.com/yd5hp9z4](https://tinyurl.com/yd5hp9z4)
SHACL

SHACL (Shapes Constraint Language)

W3C recommendation:

https://www.w3.org/TR/shacl/ (July 2017)

RDF vocabulary

2 parts: SHACL-Core, SHACL-SPARQL
Same example with blank nodes

```
prefix : <http://example.org/>
prefix sh: <http://www.w3.org/ns/shacl#>
prefix xsd: <http://www.w3.org/2001/XMLSchema#>
prefix schema: <http://schema.org/>

:UserShape a sh:NodeShape ;
    sh:targetNode :alice, :bob, :carol ;
    sh:nodeKind sh:IRI ;
    sh:property [ 
        sh:path schema:name ;
        sh:minCount 1; sh:maxCount 1 ;
        sh:datatype xsd:string ; 
    ] ;
    sh:property [ 
        sh:path schema:email ;
        sh:minCount 1; sh:maxCount 1; 
        sh:nodeKind sh:IRI ;
    ] .

:alice schema:name "Alice Cooper" ;
    schema:email <mailto:alice@mail.org> .

:carol schema:name "Carol" ;
    schema:email "carol@mail.org" .
```

Data graph

Shapes graph

Try it. RDFShape [https://goo.gl/ukY5vq](https://goo.gl/ukY5vq)
In ShEx

:AdultPerson EXTRA a {
  a [ schema:Person ] ;
  :name xsd:string ;
  :age MinInclusive 18 ;
  :gender [ :Male :Female] OR xsd:string ;
  :address @:Address ? ;
  :worksFor @:Company + ;
}

:Address CLOSED { :addressLine xsd:string {1,3} ; :postalCode /[0-9]{5}/ ; :state @:State ; :city xsd:string } ;

:Company { :name xsd:string ; :state @:State ; :employee @:AdultPerson * ; }

:State /[A-Z]{2}/

In SHACL

:AdultPerson a sh:NodeShape ;
  sh:property [ sh:path rdf:type ; sh:qualifiedValueShape [ sh:hasValue schema:Person ] ; sh:qualifiedMinCount 1 ; sh:qualifiedMaxCount 1 ] ;
  sh:qualifiedMinCount 1 ; sh:qualifiedMaxCount 1 ;
  :name xsd:string ;
  :age MinInclusive 18 ;
  :gender [ :Male :Female] OR xsd:string ;
  :address @:Address ? ;
  :worksFor @:Company + ;
}

:Address a sh:NodeShape ;
  sh:property [ sh:path :addressLine ; sh:qualifiedMinCount 1 ; sh:qualifiedMaxCount 1 ] ;

:Company a sh:NodeShape ;
  :name xsd:string ;
  :state @:State ;
  :employee @:AdultPerson * ;
}

:State a sh:NodeShape ;
  sh:pattern "[A-Z]{2}" .

Its recursive!!! (not well defined SHACL)

Implementation dependent feature

Try it: https://tinyurl.com/ycl3mkzr
Some challenges and perspectives

Theoretical foundations of ShEx/SHACL
Generating shapes from data
Validation Usability
RDF Stream validation
Schema ecosystems
  Wikidata
  Solid
Theoretical foundations of ShEx/SHACL

Conversion between ShEx and SHACL

SHacIEX library converts subsets of both

Challenges

Recursion and negation

Performance and algorithmic complexity

Detect useful subsets of the languages

Convert to SPARQL

Schema/data mapping
Generating Shapes from Data

Useful use case in practice
Knowledge Graph summarization
Some prototypes:
ShExer, RDFShape, ShapeArchitect

Try it with RDFShape:
https://tinyurl.com/y8pjcbyf
Validation usability

Learning from users

Early adopters: WebIndex, HL7 FHIR, Eclipse Lyo, GenWiki,…
Improve error information/visualization/navigation/repairing

Authoring/visualization tools

Propose annotation sets

UI generation
Error reporting/suggestion (SHOULD/MUST/…)

Shapes
RDF Stream validation

Validation of RDF streams

Challenges:
- Incremental validation
- Named graphs
- Addition/removal of triples
Schema ecosystems: Wikidata

In May, 2019, Wikidata announced ShEx adoption

New namespace for schemas


It opens lots of opportunities/challenges

Schema evolution and comparison
Schema ecosystems: Solid project

SOLID (SOcial Linked Data): Promoted by Tim Berners-Lee

Goal: Re-decentralize the Web
- Separate data from apps
- Give users more control about their data
- Internally using linked data & RDF

Shapes needed for interoperability

"...I just can’t stop thinking about shapes.", Ruben Verborgh
https://ruben.verborgh.org/blog/2019/06/17/shaping-linked-data-apps/
Conclusions

Explicit schemas (shapes) can improve linked data quality
2 languages proposed: ShEx/SHACL
Towards an ecosystem of shapes for data portals
New challenges and opportunities
More info

About ShEx

ShEx by Example (slides):
https://figshare.com/articles/ShExByExample_pptx/6291464

ShEx chapter from Validating RDF data book:

About SHACL

SHACL by example (slides):
https://figshare.com/articles/SHACL_by_example/6449645

SHACL chapter at Validating RDF data book

Comparing ShEx and SHACL
End of presentation

Acknowledgements:
David Vilches, Eridan Otto, Christian Sifaqui