“Methodology and tools for Multilingual Linguistic Linked Data generation: hands-on session”

Daniel Vila-Suero, Jorge Gracia
Ontology Engineering Group (OEG)
Universidad Politécnica de Madrid (UPM)
dvila, jgracia@fi.upm.es

EKAW Tutorial on Language Resources and Linked Data
Linköping (Sweden), 25th November 2014
Building a Linked Dataset out of Apertium Dictionaries: English lexicon

EN-ES dictionary

EN-ES RDF dictionary

Lexicon EN

Lexicon ES

Translation Set EN-ES
Goals

• Define a resource naming scheme.
• Map TSV data source to RDF re-using a set of vocabularies.
• Find and establish links to an external data source.
• Generate a (small) RDF dataset.
• Upload the RDF dataset into a triple-store
• Query the RDF dataset using SPARQL
• Install LODRefine for your platform (Win, MacOS, Linux)
Install LODRefine

- **MacOS**: *Drag and drop to Applications folder*

- **Windows**:
  - Uncompress it into a folder wherever you want (such as in C:\LODrefine).

- **Linux**:
  - Execute:
    
    ```
    tar xzf lodrefine.1.0.7.tar.gz
    cd lodrefine.1.0.7
    ```
Add RDF Extension

• Place the rdf-extension folder in:
  • Windows and Linux:
    – webapp/extensions folder
  • MacOS:
    – Resources/webapp/extensions folder

• Run LOD Refine
Choose file: data/raw-data-apertium-en-es_lexiconEN
Create project

Character encoding: UTF8
Task: Mapping file to RDF using lemon

1. Select Edit RDF Skeleton
2. Set your base URI
3. Add prefix of lemon:
   - http://lemon-model.net/lemon#
4. Start creating the mapping (RDF Schema alignment)
Editing the RDF Skeleton

RDF Schema Alignment

The RDF schema alignment skeleton below specifies how the RDF data that will get generated from your grid-shaped data. The cells in each record of your data will get placed into nodes within the skeleton. Configure the skeleton by specifying which column to substitute into which node.

Base URI: http://rdf.sporium.org/id/edit

Available Prefixes: rdfs foaf owl xsd rdf lemon + add prefix + manage prefixes

Check results

Create instances and add a type

Define object

Create Properties (literal and objects)

25/11/2014

Daniel Vila-Suero, Jorge Gracia
Editing the RDF Skeleton

RDF Node

- Use content from cell...
  - (row index)
  - LexicalEntry-id
  - LexicalEntry-POS
  - LexicalEntry-writtenForm
  - sense
  - Constant Value

- The cell's content is used...
  - as a URI
  - Create instances and object properties range
  - as text
  - as language-tagged text
  - Literals and language-tagged literals
  - as integer number
  - as non-integer number
  - as date (YYYY-MM-DD)
  - as dateTime (YYYY-MM-DD HH:MM:SS)
  - as boolean
  - as custom datatype (specify type URI)
  - as a blank node

- Use custom expression...
  - value
  - preview/edit

- Apply transformations using GREL (GR expression language)
1. Create a reconciliation service from RDF file (data/linking-dataset-rdf-wordnet-extract.nt)

2. Duplicate *LexicalEntry-writtenForm* column

3. Reconcile using created service (e.g. *wordnet*)
Creating a reconciliation service

25794 rows

<table>
<thead>
<tr>
<th>No.</th>
<th>LexicalEntry-id</th>
<th>LexicalEntry-PO</th>
<th>wordnet-pos</th>
<th>sense</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>object-adj-en</td>
<td>adj</td>
<td><a href="http://wordnet-princeton.edu">http://wordnet-princeton.edu</a></td>
<td></td>
</tr>
<tr>
<td>2.</td>
<td>abrupt-adj-en</td>
<td>adj</td>
<td><a href="http://wordnet-princeton.edu">http://wordnet-princeton.edu</a></td>
<td></td>
</tr>
<tr>
<td>3.</td>
<td>beach-word-verb</td>
<td>vb lex</td>
<td><a href="http://wordnet-princeton.edu">http://wordnet-princeton.edu</a></td>
<td></td>
</tr>
<tr>
<td>4.</td>
<td>startfresh-verb</td>
<td>vb lex</td>
<td><a href="http://wordnet-princeton.edu">http://wordnet-princeton.edu</a></td>
<td></td>
</tr>
<tr>
<td>5.</td>
<td>abides-by-verb</td>
<td>vb lex</td>
<td><a href="http://wordnet-princeton.edu">http://wordnet-princeton.edu</a></td>
<td></td>
</tr>
<tr>
<td>6.</td>
<td>seaurchin-noun</td>
<td>n</td>
<td><a href="http://wordnet-princeton.edu">http://wordnet-princeton.edu</a></td>
<td></td>
</tr>
<tr>
<td>7.</td>
<td>abode-noun</td>
<td>n</td>
<td><a href="http://wordnet-princeton.edu">http://wordnet-princeton.edu</a></td>
<td></td>
</tr>
<tr>
<td>8.</td>
<td>abruptness-noun</td>
<td>n</td>
<td><a href="http://wordnet-princeton.edu">http://wordnet-princeton.edu</a></td>
<td></td>
</tr>
<tr>
<td>9.</td>
<td>tradeunion-noun</td>
<td>n</td>
<td><a href="http://wordnet-princeton.edu">http://wordnet-princeton.edu</a></td>
<td></td>
</tr>
<tr>
<td>10.</td>
<td>tradeunion-noun</td>
<td>n</td>
<td><a href="http://wordnet-princeton.edu">http://wordnet-princeton.edu</a></td>
<td></td>
</tr>
</tbody>
</table>

Based on SPARQL endpoint...
Based on RDF file...
Based on a Sindice site search...
Based on a Apache Starbol EntityHub.
Creating a reconciliation service

Add file-based reconciliation service

This will set up a new reconciliation service based on an RDF file that provides entity URIs and entity labels.

Name: wordnet
A human readable name

File details
- Load file from URL:
- Upload file: Seleccionar archivo
  Ningún archivo seleccionado
File format: N-Triple

Label properties
Select properties that are used to label resources in the RDF data. These properties will be used to match resources:
- rdfs:label
- skos:prefLabel
- dcterms:title
- dc:title
- foaf:name
- Other...

OK Cancel
Reconciling a column

25794 rows

Show as: rows records Show: 5 10 25 50 rows

Reconcile
Reconciling a column

Reconcile against type: http://wordnet-rdf.princeton.edu/ontology#Synset
Validate and map the matches to RDF

Use Reconciliation facets and filter:
E.g. Best candidate's score

Validate!
Validate and map the matches to RDF

1. The URI to external source is available using GREL:
   
   `cell.recon.match.id`

2. Use RDF Skeleton and match using your preferred linking property:
   
   e.g. `owl:sameAs, skos:closeMatch, lemon:reference`
1. Export to file:

**RDF as Turtle menu option**
Task: Loading RDF data and querying

1. Launch Fuseki triplestore
2. Load data
3. Query
Launching Fuseki

- Execute:
  
  ```bash
  fuseki-server --update --mem /handson
  
  Creates a fuseki server storing data in memory with rights to update data
  ```

- Server accessible under:

  `http://localhost:3030/`
SPARQL Query

Output: Text
If XML output, add XSLT style sheet (blank for none):
Force the accept header to text/plain regardless.
Get Results

SPARQL Update

Perform update

File upload

File: Ningún archivo seleccionado
Graph: default