Explore to Understand: Enriched Semantic Web Application

Project - Web of Data

Joe Raad

Context

In this project, you will apply your skills in semantic web technologies:

- creation and manipulation of RDF graphs,
- definition and use of ontologies,
- writing SPARQL queries,
- enrichment with external data (Wikidata).

The goal of this project is to introduce you to the complete implementation of a semantic web application. Starting from an open French dataset, you will:

- formulate a clear and motivated research question,
- transform the dataset into an RDF graph by defining an appropriate ontology,
- enrich your local data with complementary information from Wikidata,
- query and analyze the constructed knowledge graph,
- design a web application that visualizes your results clearly and instructively.

You will work in groups of two and must deliver a reproducible project including the CSV dataset, the enriched RDF graph, your SPARQL queries, the code of your web application, and an execution guide (README). The whole project will be accompanied by a short oral presentation intended for your classmates, explaining the context, the chosen question, and the visualization of your results.

Project Steps

1. Dataset selection and exploration

Choose a French dataset (for example data.gouv.fr, INSEE, Etalab Platforms).

Questions to consider:

1. What interesting question would you like to ask about this dataset?
    - **Examples (political questions):** How many local elected officials hold multiple positions (mayor, MP, regional councillor), and how does this differ across regions? Which departments have the highest percentage of female elected officials (mayors, regional councillors), and has this percentage improved over the past 20 years? Which countries maintain the most twinning partnerships with French cities, and how are these partnerships distributed by continent?
    - **Examples (social questions):** Which departments combine low medical density with an aging population? Is there a correlation between high school success rates and median income across municipalities?
2. What complementary information from Wikidata could enrich your dataset?
    - Images of people or places
    - Precise geographic coordinates
    - Dates of birth or foundation
    - Affiliations or entity types
    - Any other information relevant to your question that are not in the CSV dataset

2. Conversion to RDF and import into GraphDB

- Define an appropriate ontology with Protégé.
- Convert the tabular dataset into RDF (using OntoRefine or another tool).
- Create a GraphDB repository and import your RDF data.

3. Enrichment with Wikidata

- Write federated SPARQL queries to link your local entities to those in Wikidata (`owl:sameAs`).
- Use CONSTRUCT queries to test the links, then INSERT queries to enrich your graph.
- Example of information to add: images, coordinates, affiliations, categories.

At the end of this step: your graph should contain both the `owl:sameAs` links between your local entities and Wikidata entities, as well as the enriched information added directly to your local entities. Check that all relevant entities are linked to a corresponding entity in Wikidata. If some are not, analyze why (different names, missing data, ambiguities, etc.) and adjust your linking queries accordingly.

4. Query and analysis

- Write one or more SPARQL queries that answer the research question defined in step 1.
- Analyze the results carefully:
  - Do the results fully answer your question?
  - What limitations or gaps do you observe?
  - How did enrichment with Wikidata improve the answer?
- Provide a clear and well-argued interpretation: it is not enough to execute a query, you must demonstrate how the results provide a meaningful answer to your initial question.

5. Visualization and web application

Develop a web application connected to the local SPARQL endpoint of GraphDB.

You are free to choose the language and framework for your application (Python, JavaScript, etc.). The visualization must be adapted to the data and the research question. Some suggested libraries:

- Interactive maps: Leaflet.js (JavaScript), Folium (Python)
- Charts and graphs: Chart.js, D3.js (JavaScript), matplotlib, plotly (Python)
- Timelines: vis.js, plotly
- Image galleries: Lightbox, Bootstrap Carousel

Deliverables

You must submit on eCampus a **compressed folder (.zip)** containing:

1. The original dataset CSV file (+ a description of any modifications).
2. The SPARQL query files (.rq).
3. The enriched RDF file (.ttl) after linking with Wikidata.
4. The source code of your web application (or a GitHub link).
5. A `README` file (`.txt`, `.md`, or `.pdf`) serving as a reproduction guide:
   - Describe all steps needed to reproduce your project (installation, configuration, data import, application launch).
   - Include all necessary links (dataset, source code, dependencies).
   - Specify any constraints (e.g., specific GraphDB repository name, tool versions, etc.).
   - Test that your project can be fully reproduced on a clean machine following only this guide.

6. A short oral presentation (5 minutes):
    - Present the context, research question, and methodological choices.
    - Show your visualization and results.
    - The goal is to explain your approach clearly to your classmates without delving into every technical detail.

Grading: 20 points

1. Dataset choice and question (6 points)

- Relevance of the dataset (richness, usefulness, size)
- Research question (clear, motivated, original)
- Quality of Wikidata complementary data (real added value for analysis)

2. Modeling and RDF (3 points)

- Well-defined ontology (appropriate classes and properties)
- Use of a reasoner for relevant inferences
- Correct and complete RDF conversion

3. Wikidata enrichment (3 points)

- Well-established `owl:sameAs` links
- Relevant information retrieved and integrated

4. Final SPARQL query and analysis (3 points)

- Correct and optimized query
- Coherent and usable results
- Clear analysis and meaningful interpretation

5. Web application and visualization (3 points)

- SPARQL connection and dynamic retrieval
- Visualization adapted to data type
- Quality and usability of the interface

6. Oral presentation and deliverables (2 points)

- Clarity and structure of the oral presentation
- Quality of the demonstration and pedagogical approach
- Quality of deliverables: documentation and files allowing full project reproduction