Task 3

Overview

Using “NobelLaureates KG” (in Turtle) describing Nobel prize winners, and a “scientistsBio.ttl” with biographical details from Wikidata the task is to:

1. Merge individuals who appear in both datasets into a new KG, updatedKG.ttl.
2. Query the merged or original dataset(s) to answer three specific SPARQL queries about Nobel laureates.

---

1. Construct Query

This CONSTRUCT query merges biographical data from the scientists’ biographies (scientistsBio.ttl) with Nobel prize information (NobelLaureatesKG.ttl). It combines data for individuals found in both graphs based on matching their Wikidata URIs (using ***owl:sameAs).***

The merged dataset includes:

• Unified names (***foaf:name***)

• Birth details (***dbpedia-owl:birthDate, dbpedia-owl:birthPlace***)

• Nobel Prize details ***(nobel:laureateAward, nobel:nobelPrize, nobel:motivation***, prize labels (***rdfs:label***), and award dates (***dcterms:dat***

SPARQL Construct Query

```
import rdflib
from rdflib.namespace import OWL

# Load RDF datasets
nobel_graph = rdflib.Graph()
bio_graph = rdflib.Graph()
nobel_graph.parse("NobelLaureatesKG.ttl", format="turtle")
bio_graph.parse("scientistsBio.ttl", format="turtle")

# Identify common individuals using owl:sameAs (Wikidata URIs)
bio_uris = set(bio_graph.subjects())
common_uris = set()
```

```
for subj, obj in nobel_graph.subject_objects(predicate=OWL.sameAs):
    if isinstance(obj, rdflib.URIRef) and obj in bio_uris:
        common_uris.add(obj)

# Construct merged knowledge graph
merged_graph = rdflib.Graph()

for uri in common_uris:
    # Add biographical information from scientistsBio
    for p, o in bio_graph.predicate_objects(subject=uri):
        merged_graph.add((uri, p, o))

    # Add Nobel Prize details from NobelLaureatesKG
    for laureate in nobel_graph.subjects(predicate=OWL.sameAs, object=uri):
        for p, o in nobel_graph.predicate_objects(subject=laureate):
            merged_graph.add((laureate, p, o))
        # Include owl:sameAs link between laureate and Wikidata URI
        merged_graph.add((laureate, OWL.sameAs, uri))

# Save merged graph to updatedKG.ttl
merged_graph.serialize("updatedKG.ttl", format="turtle")
```

1. The script reads the original NobelLaureatesKG.ttl and scientistsBio.ttl LRDF graphs.
2. Using the owl:sameAs property, the script finds common Wikidata URIs that appear in both datasets.
3. Merge the data: For each matched individual:
    - Biographical information (birth date, birthplace, names) is added from scientistsBio.ttl.
    - Nobel prize details (awards, motivations, prizes) are integrated from NobelLaureatesKG.ttl.
    - owl:sameAs relationships are explicitly retained to link Nobel laureate URIs with Wikidata URIs.
4. Output: Finally, the combined knowledge graph (updatedKG.ttl) is serialised in Turtle syntax, containing complete merged data

Query #1: Individuals Present in Both Datasets

To list all individuals that appear in both the Nobel KG and scientistsBio KG using owl:sameAs links.

SPARQL Query

```
query1 = """
    PREFIX ns1: <http://data.nobelprize.org/terms/>
    PREFIX foaf: <http://xmlns.com/foaf/0.1/>
    PREFIX owl: <http://www.w3.org/2002/07/owl#>
    SELECT DISTINCT ?name WHERE {
      ?laur a ns1:Laureate ;
            owl:sameAs ?wd .
      ?wd foaf:name ?name .
    }
    ORDER BY ?name
    """
    results1 = merged_graph.query(query1)
    print("Query 1: Individuals present in both datasets:")
    for row in results1:
        print(f" - {row.name}")
    print()
```

• Retrieves laureate resources in the Nobel KG linked via ***owl:sameAs*** to a Wikidata URI.

• Uses that URI to extract the ***foaf:name*** from the scientistsBio KG.

• Returns a distinct, alphabetically ordered list of names.

Output

- Andre Geim
- Aziz Sancar
- Benjamin List
- Camillo Golgi
- Charles K. Kao
- Emmanuelle Charpentier
- Finn E. Kydland
- George E. Smith
- Giorgio Parisi
- Guglielmo Marconi
- Heike Kamerlingh Onnes

- Ivar Giaever
- John O’Keefe
- Julius Axelrod
- Maria Goeppert Mayer
- Martin Karplus
- Max Born
- Niels Ryberg Finsen
- Peter Debye
- Reinhard Genzel
- Rita Levi-Montalcini
- Serge Haroche
- Sheldon Glashow
- Susumu Tonegawa

Query #2: Nobel Laureates Born in France, Chemistry, 1 Co-Laureate

Identify laureates who:

• Are born in France (using ***dbo:birthPlac***e).

• Won a Chemistry Nobel Prize.

• Shared the prize with exactly one other laureate.

SPARQL Query

```
query2 = """
  PREFIX ns1: <http://data.nobelprize.org/terms/>
  PREFIX foaf: <http://xmlns.com/foaf/0.1/>
  PREFIX dbo: <http://dbpedia.org/ontology/>
  SELECT DISTINCT ?name WHERE {
    ?laur a ns1:Laureate ;
          foaf:name ?name ;
          dbo:birthPlace <http://data.nobelprize.org/resource/country/France> ;
          ns1:nobelPrize ?np .
    ?np ns1:category ns1:Chemistry .
    ?co a ns1:Laureate ;
        ns1:nobelPrize ?np .
    FILTER(?co != ?laur)
    FILTER NOT EXISTS {
        ?other a ns1:Laureate ;
               ns1:nobelPrize ?np .
        FILTER(?other != ?laur && ?other != ?co)
    }
  }
  ORDER BY ?name
  """
  results2 = merged_graph.query(query2)
  print("Query 2: Nobel Laureates born in France who won the Chemistry prize with exactly one co-laureate:")
  for row in results2:
        print(f" - {row.name}")
  print()
```

The subquery first identifies all Chemistry prizes with exactly 2 laureates.

• Then, for those prizes, it returns the names of the laureates who satisfy the birthPlace condition.

Output

- Emmanuelle Charpentier
- Frédéric Joliot
- Irène Joliot-Curie
- Moungi Bawendi
- Paul Sabatier
- Victor Grignard

Query #3: Laureates with More Than One Nobel Prize

To list each laureate who has won more than one Nobel Prize, along with each prize’s URI, label, and awarding year.

SPARQL Query

```
query3 = """
    PREFIX ns1: <http://data.nobelprize.org/terms/>
    PREFIX foaf: <http://xmlns.com/foaf/0.1/>
    PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
    SELECT ?name (GROUP_CONCAT(CONCAT(?categoryName, " (", STR(?year), ")"); separator=", ") AS
?prizes)
    WHERE {
      ?laureate a ns1:Laureate ;
                foaf:name ?name ;
                ns1:nobelPrize ?prize .
```

```
  ?prize ns1:category ?cat ;
         ns1:year ?year .
  # Convert the category URI into a readable name
  BIND( REPLACE(REPLACE(STR(?cat), "^.*/", ""), "_", " ") AS ?categoryName )
}
GROUP BY ?laureate ?name
HAVING (COUNT(DISTINCT ?prize) > 1)
ORDER BY ?name
"""
results3 = merged_graph.query(query3)
print("Query 3: Nobel Laureates who won more than one Nobel Prize (with years and categories):")
for row in results3:
    print(f" - {row.name}: {row.prizes}")
print()
```

- Groups laureate by name and count distinct prize entries.
- Uses GROUP_CONCAT to display prize details (year and category) in a single string.

Output

- Frederick Sanger: Chemistry (1958), Chemistry (1980)
- John Bardeen: Physics (1956), Physics (1972)
- K. Barry Sharpless: Chemistry (2001), Chemistry (2022)
- Linus Pauling: Chemistry (1954), Peace (1962)
- Marie Curie: Physics (1903), Chemistry (1911)
- International Committee of the Red Cross: Peace (1917), Peace (1944), Peace (1963)
- Office of the United Nations High Commissioner for Refugees: Peace (1954), Peace (1981)
- Den internasjonale Røde Kors-komiteen: Peace (1917), Peace (1944), Peace (1963)
- FNs høykommissariat for flyktninger: Peace (1954), Peace (1981)
- FN:s flyktingkommissariat: Peace (1954), Peace (1981)
- Internationella Rödakorskommittén: Peace (1917), Peace (1944), Peace (1963)

Query #4: Laureates Affiliated with German Universities, Winning Physics Prize

Find Physics laureates with affiliations (using ***dbo:affiliation***) to German universities.

SPARQL Query

```
query4 = """
  PREFIX ns1: <http://data.nobelprize.org/terms/>
  PREFIX foaf: <http://xmlns.com/foaf/0.1/>
  PREFIX dbo: <http://dbpedia.org/ontology/>
  SELECT DISTINCT ?name WHERE {
    ?laureate a ns1:Laureate ;
              foaf:name ?name ;
              ns1:nobelPrize ?np ;
              dbo:affiliation ?univ .
    ?np ns1:category ns1:Physics .
    ?univ dbo:country <http://data.nobelprize.org/resource/country/Germany> .
  }
  ORDER BY ?name
  """
  results4 = merged_graph.query(query4)
  print("Query 4: Nobel Laureates (Physics) affiliated with universities in Germany:")
  for row in results4:
      print(f" - {row.name}")
  print()
```

- Retrieves laureates whose prize category is Physics.
- Filters for affiliations where the associated university's country is Germany.

Output

- Albert Einstein
- Ernst Ruska
- Erwin Schrödinger
- Ferdinand Braun
- Ferenc Krausz
- Gustav Hertz
- J. Hans D. Jensen
- James Franck
- Johannes Stark
- Klaus Hasselmann

- Klaus von Klitzing
- Max Planck
- Max von Laue
- Peter Grünberg
- Philipp Lenard
- Reinhard Genzel
- Rudolf Mössbauer
- Theodor W. Hänsch
- Walther Bothe
- Werner Heisenberg
- Wilhelm Conrad Röntgen
- Wilhelm Wien
- Wolfgang Paul