Knowledge Retrieval Seminar Work

2300341081
Gwang Won Seo

1. About Dataset
2. Using Tools
3. Procedure
4. Limitation

gitlab: https://gitlab.web.fh-kufstein.ac.at/seo_gwang_won/python-sparql

1.About Dataset

2 RDF Datasets were downloaded from KBpedia which are ‘SocialSystems-typology.n3’ and ‘Society-typology.n3’.

2.Using Tools

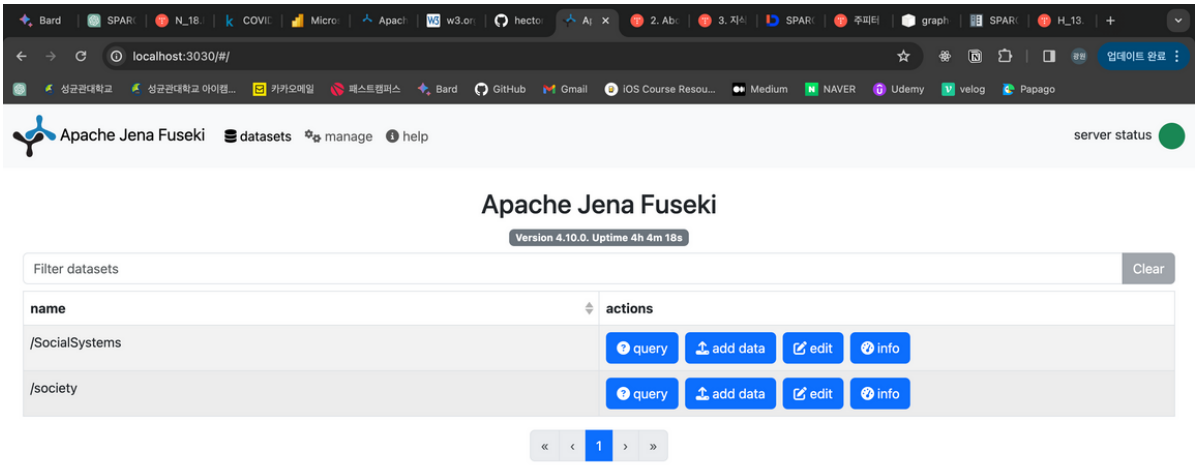
- Python
: It supports multiple programming paradigms, has a vast standard library, and is widely used in web development, data science, AI, and automation.

- Jupyter Notebook
: It is a web-based tool for interactive coding and documentation. It lets you combine live code, visualizations, and text in a collaborative environment, commonly used in data science with a focus on Python.

- Apache Jena Fuseki
: It is a helpful tool for handling and accessing RDF data on the web, making it easier for computers to understand and process certain types of information.

3.Procedure

1) Store data using Apache Jena Fuseki



2) Query using python

```python
# Question : Let me know all living languages.

# Set the Query
sparql_society.setQuery("""
    PREFIX kko: <http://kbpedia.org/kko/rc/>
    PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

    SELECT ?languages
    WHERE {
        ?languages rdfs:subClassOf kko:LivingLanguage.
}
""")

# Specify the format of the results (JSON in this case)
sparql_society.setReturnFormat(JSON)

# Execute the query and get the results
results = sparql_society.query().convert()

# Print the results
for result in results["results"]["bindings"]:
    language = result["languages"]["value"]
    print(language[26:])
```

```
ModernGreekLanguage
NootkaLanguage
MarshalleseLanguage
WelshLanguage
IcelandicLanguage
ShillukLanguage
EgyptianArabicLanguage
SwahiliLanguage
NandiLanguage
KhasiLanguage
CocopaLanguage
```

```python
# Question: What are included in Public Policies?

# Set the SPARQL query
sparql_ss.setQuery("""
    PREFIX kko: <http://kbpedia.org/kko/rc/>
    PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

    SELECT ?publicpolicies
    WHERE {
        ?publicpolicies rdfs:subClassOf kko:PublicPolicy
    }
""")

# Specify the format of the results (JSON in this case)
sparql_ss.setReturnFormat(JSON)

# Execute the query and get the results
results = sparql_ss.query().convert()

# Print the results
for result in results["results"]["bindings"]:
    publicpolicy = result["publicpolicies"]["value"]
    print(publicpolicy[26:])
```

```
EnergyPolicy
HealthPolicy
NuclearPolicy
EducationPolicy
PublicPolicyAddressingHomelessness
RenewableEnergyPolicy
TradePolicy
EconomicPolicy
SpacePolicy
IndustrialPolicy
MacroeconomicPolicy
MonetaryPolicy
EnvironmentalPolicy
```

4.Limitation

- I didn’t fully understand the datasets, so couldn’t make in-depth questions.