Semantic Data Lake

- Members:
  1. Sayed Hoseini
  2. Tobias Claas
  3. Maher Fallouh
  4. Abdullah Zaid
  5. Muhammad Noman

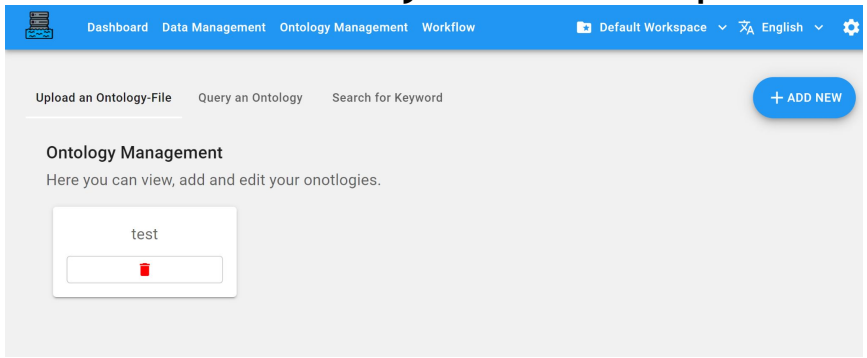
Introduction

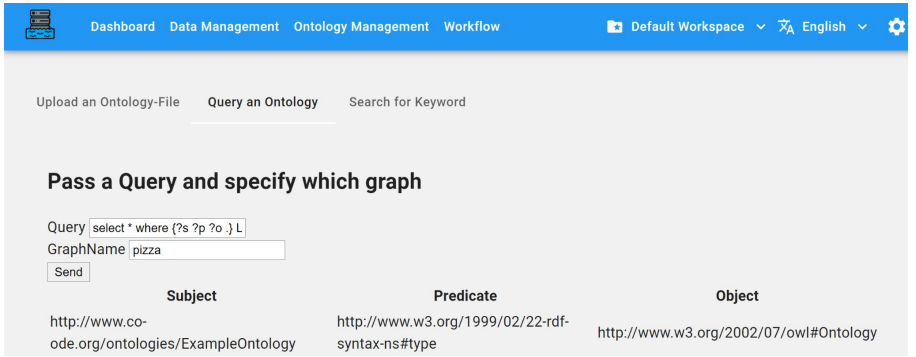
Main Contributions so far:

- New FrontEnd written in TS/React
- WorkSpace API
- Ontology Management with Fuseki
- Extracted an Ontology from National Cancer Institute Thesaurus
- Workflow Diagram with ReactFlow
- API for Data Transformation executed in Spark Cluster (Proof-of-Concept)

Fuseki

- Create a Fuseki-DataBase for every Workspace
- Ingest Ontology-Data (.owl, .rdf, .n3, …) as named Graphs
- Query for a specific Graph or generic Querystring
- Search for Keywords in Graph or the complete Ontology





- Next Steps:

  Create connections between Datasets and Ontology. We will Use MongoDB / Flask Documents, i.e. JSONs to store the information.

Workspace

- Workspaces have separate ontologies and data
  - New Fuseki, Hadoop, MongoDB, Postgres dataset for every workspace
- We will use NCIT as the Standard Ontologie (Property or Attribute)
  - Others can be added and managed in section “Manage Ontologies”

Ontologie - NCIT

- National Cancer Institute Thesaurus
  - Property or Attribute subset
    - Property or Attribute Class
    - Subclasses
    - Labels for referenced classes that are not Subset of ‘PoA’
- Query
  - Property Path
  - Optimized version

```
top - 12:44:03 up 3 days,  3:18,  6 users,  load average: 1.35, 1.15, 1.08
Tasks: 107 total,   3 running, 104 sleeping,   0 stopped,   0 zombie
%Cpu(s): 98.8 us,  1.2 sy,  0.0 ni,  0.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem :  64230.7 total,  57910.2 free,   4200.4 used,   2120.1 buff/cache
MiB Swap:      0.0 total,      0.0 free,      0.0 used.  59463.7 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
  28052 sayed     20   0  136788 127012   8668 R  99.7   0.2   0:22.75 python3
  24798 sayed     20   0 3807692   3.6g   9060 R  98.7   5.8 413:40.16 python3
  27956 sayed     20   0 1046156  92664  34696 S   0.7   0.1   0:03.37 node
```

```
CONSTRUCT {
    ?poa ?a ?b .
    ?a rdfs:label ?l1 .
    ?b rdfs:label ?l2 .
} WHERE {
    {
        ?poa (rdfs:subClassOf)* ?x ;
            ?a ?b .
        OPTIONAL {?a rdfs:label ?l1 .}
        OPTIONAL {?b rdfs:label ?l2 .}
        FILTER(?x = ncid:C20189)
    }
}""")
```

Ontologie - NCIT

- From ~600 MB to 6 MB
  - So querying for keywords can be fast
- Found Dataset
  - Annotated some of the columns with NCIT PoA vocabulary
  - Steps:
    - Looked up Description of the columns
    - Searched for matching attribute on NCITs
      - Made sure it is in PoA subset
    - Mapping between columns in dataset and attributes in Ontologie in JSON format
      - To be inserted into MongoDB
  - Searching in Web-App for Ontologie Attributes

Workflow - Backend

- Work done so far:
  - CSV ingestion to HDFS
  - Get all ingested datamart API
  - Transformation API for SELECT and JOIN
  - Standard data structure: Datamart (Reading/Writing data)
  - Loading pyspark dataframe from datamart

- Future work:
  - Mongo, postgres ingestion
  - Other transformations like groupby, filter
  - Implementation of jobs

Workflow - Backend

- Translating frontend workflow into Job tasks.
- Job tasks consists:
  - Data ingestion
  - Data transformation i.e. Join, select, filter
  - Data persisting at end of workflow -> Datamart

Next Steps

- Create connections between Datasets and Biology Ontology. We will Use MongoDB / Flask Documents, i.e. JSONs to store the information.
- Implement a Job Abstraction to translate WorkFlow Diagram into Tasks that run on the Spark Cluster
- Add more Data Transformations
  - Mongo, postgres ingestion
  - GroupBy, Filter, ...