SPARQL Queries

No Author Given

No Institute Given

We present here 2 queries that gives an overview of the possibles manipulations of entity matching contexts within an RDF graph. The knowledge graph build for this use case results of the by merging DBpedia and YAGO entities of the class Museum and the addition of their entity matching contexts described according the RDF vocabulary provided in the article. The first query shows EMCs can be requested from its properties. For example we can list all pairs of entities in a the entity matching context that includes the property $\{islocatedin\}$ in ε, $\{preflabel, wascreatedonyear\}$ in $\mathit{\Delta}$ and $\{haslatitude, haslongitude, wascreatedondate\}$ in $\mathit{\Omega}$:

```
SELECT  ?x ?y ?emc
 WHERE {
     ?x ?emc ?y .

     ?emc a ns1:EntityMatchingContext ; ns1:epsilon
         ?epsilon ; ns1:delta  ?delta ; ns1:omega
        ?omega .

     ?epsilon ns1:includes ns1:islocatedin .
     #?epsilon ns1:size 1 .

     ?delta ns1:includes ns2:preflabel ; ns1:
        includes ns1:wascreatedonyear .
     #?delta ns1:size 2 .

     ?omega ns1:includes ns1:haslatitude , ns1:
        haslongitude , ns1:wascreatedondate .
     #?omega ns1:size 3 .
     }
```

Notice that if no sizes are given, we obtain not only the contexts specified, but also all the above contexts of the ε, $\mathit{\Delta}$ and $\mathit{\Omega}$ lattices (i.e. contexts that are more specific than those queried).

The second query shows that given a pair of entities we can obtain the properties of their EMCs. For example, we list the properties of the entity matching context of the pair: ”appalachian trail” museum for YAGO and ”the museum of the american revolution” from DBpedia.

```
  SELECT ?eprop ?dprop ?oprop
 WHERE {
```

```
<appalachian_trail_museum_yago> ?p <
   museum_of_the_american_revolution_db> .
?p a ns1:EntityMatchingContext ;
    ns1:epsilon/ns1:includes ?eProp ;
    ns1:delta/ns1:includes ?dProp ;
    ns1:omega/ns1:includes ?oProp .
}
```

The EMCs for this pair is: $\{islocatedin\}, \{islocatedin, skos : preflabel\}, \{wascreatedonyear\}$

Code: The generation of the rdf graph is done by the script: iswc2024/rdf/museum_rdf.py. Queries are in the script: iswc2024/rdf/test_request.py. Queries results are in files iswc2024/rdf/pair_with_prop.res and swc2024/rdf/emc_with_prop.res.