Meta-sparql examples

Document draft version from: 2017-07-24
Author: Johannes Frey (AKSW/KILT, Leipzig University)
Feedback / Comments: https://github.com/AKSW/meta-sparql/issues/new (tag as enhancement or question)

Contents

Introduction

In order to enable MRM-independent SPARQL queries, the generic and extensible tool meta-SPARQL has been developed. It allows automatic rewriting of SPARQL queries for different MRMs. The idea is, to replace every triple pattern within a SPARQL query by a set of special annotations, which will be translated by meta-SPARQL into the appropriate format. Every query needs to be written as a template in an intermediate SPARQL dialect based on these annotations. It consists of 4 annotations explained in Table 1. The template can be converted into query instances of the various MRMs. Therefore query templates can be written independent of granularity support and other MRM-specific characteristics. The semantics of every annotation is further illustrated by a set of examples online2. Meta-SPARQL features a file format to convert several query templates at once, which can be used for MRM benchmarking purposes, but also exposes functions for the conversion of single queries or annotations.

meta-rdf query annotations

In order to translate the queries, we developed a generic tool for rewriting SPARQL queries for different MRMs.
The idea is that triple patterns within a query are replaced by special annotation, which will be translated into the appropriate format.
It consists of the following annotations:

#!data(?s,?p,?o)!#	accessing a regular data triple (needed for regular data queries)
#!reif(?id,?s,?p,?o)!#	analogous to #!data but retrieving statement id as well
#meta(?id,?k,?v)!#	retrieve metadata key and value given statement id
#meta2(?id,?k,?v)!#	retrieve metadata key and value, which is reified itself (due to meta-metadata), given a statement id
|x|	x denotes a template variable, which gets replaced by a specific constant to derive query instances from the template

to illustrate the different behaviour, an example translation of the annotations into standard reification MRM is shown below:

example annotation translation

#!data(?s,?p,?o)!#

```
?dummyVar_0 a <http://www.w3.org/1999/02/22-rdf-syntax-ns#Statement>;
<http://www.w3.org/1999/02/22-rdf-syntax-ns#subject>  ?s;
<http://www.w3.org/1999/02/22-rdf-syntax-ns#predicate>      ?p;
<http://www.w3.org/1999/02/22-rdf-syntax-ns#object>         ?o.
```

#!reif(?id,?s,?p,?o)!#

```
?id a <http://www.w3.org/1999/02/22-rdf-syntax-ns#Statement>;
<http://www.w3.org/1999/02/22-rdf-syntax-ns#subject>  ?s;
<http://www.w3.org/1999/02/22-rdf-syntax-ns#predicate>      ?p;
<http://www.w3.org/1999/02/22-rdf-syntax-ns#object>         ?o.
```

#meta(?id,?k,?v)!#

```
?id <http://sdw.aksw.org/metardf/hasSharedMeta> ?shared_2 .?shared_2 ?k ?v .
```

#meta2(?id,?k,?v)!#

```
?id <http://sdw.aksw.org/metardf/hasSharedMeta> ?shared_3 .?dummyVar_3 a
<http://www.w3.org/1999/02/22-rdf-syntax-ns#Statement>;
<http://www.w3.org/1999/02/22-rdf-syntax-ns#subject>  ?shared_3;
<http://www.w3.org/1999/02/22-rdf-syntax-ns#predicate>      ?k;
<http://www.w3.org/1999/02/22-rdf-syntax-ns#object>         ?v.
```

Example (data-only) query templates

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%    DBQ-SIM-01
SELECT ?p ?o
WHERE {
      #!data(<|person|>,?p,?o)!#
}
LIMIT 1000

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%    DBQ-SIM-02
PREFIX dbo: <http://dbpedia.org/ontology/>
SELECT DISTINCT ?city ?pop
WHERE {
      #!data(?city,dbo:populationTotal,?pop)!#
      #!data(?city,dbo:country,<|country|>)!#
      FILTER(?pop>20000||?pop>"20000")
}
LIMIT 10000
```

Mixed (data and metadata) example query templates

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%    DBM-SIM-01
SELECT ?p ?o ?date
WHERE {
      #!reif(?id,<|person|>,?p,?o)!#
      #!meta2(?id,<http://purl.org/dc/element/1.1/created>,?date)!#
}
LIMIT 1000

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%    DBM-SIM-02
PREFIX dbo: <http://dbpedia.org/ontology/>
SELECT DISTINCT ?city ?pop ?provenance
WHERE {
      #!reif(?id,?city,dbo:populationTotal,?pop)!#
      #!data(?city,dbo:country,<|country|>)!#
      #!meta(?id,<http://ns.inria.fr/dbpediafr/voc#hasMainRevision>,?provenance)!#
      FILTER(?pop>20000||?pop>"20000")
}
LIMIT 10000

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%    DBM-MED-01
PREFIX owl: <http://www.w3.org/2002/07/owl#>
SELECT ?p ?o ?confidence
WHERE {
      #!reif(?id,<|e_en|>,owl:sameAs,?e_de)!#
      #!data(<|e_en|>,?p,?o)!#
      FILTER EXISTS { #!data(?e_de,?p,?o)!# }
      #!meta2(?id,<http://ns.inria.fr/dbpediafr/voc#uniqueContributorNb>,?confidence)!#
      FILTER(?e_de!=<|e_en|>).
}

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%    DBM-MED-02
PREFIX dbo: <http://dbpedia.org/ontology/>
SELECT (count(distinct ?company) as ?c)
WHERE {
      #!reif(?id2,?company,dbo:locationCountry,<|country|>)!#
      #!reif(?id,?company,rdf:type,dbo:Company)!#
```

```
      #!meta2(?id,<http://ns.inria.fr/dbpediafr/voc#uniqueContributorNb>,?cont)!#
      #!meta2(?id2,<http://ns.inria.fr/dbpediafr/voc#revPerYear2016>,?revs)!#
      FILTER(?revs >5 && ?cont>10)
}

## %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%    DBM-HAR-01
PREFIX dbo: <http://dbpedia.org/ontology/>
SELECT  DISTINCT ?company1 ?mod1 ?mod2
WHERE {
            #!data(?company1,rdf:type,dbo:Company)!#
            #!data(?company2,rdf:type,dbo:Company)!#
            #!reif(?id1,?company1,dbo:industry,<|sector|>)!#
            #!meta2(?id1,<http://purl.org/dc/element/1.1/modified>,?mod1)!#
            #!reif(?id2,?company2,dbo:industry,<|sector|>)!#
            #!meta2(?id2,<http://purl.org/dc/element/1.1/modified>,?mod2)!#
          OPTIONAL{
            #!data(?company1,rdfs:label,?label1)!#
        } OPTIONAL{
            #!data(?company2,rdfs:label,?label2)!#
            } OPTIONAL{
            #!data(?company1,dbo:locationCity,?city1)!#
        } OPTIONAL{
            #!data(?company2,dbo:locationCity,?city2)!#
            } OPTIONAL{
            #!data(?company1,dbo:locationCountry,?country1)!#
        } OPTIONAL{
            #!data(?company2,dbo:locationCountry,?country2)!#
            }
            FILTER (?company1!=?company2)
        FILTER ( (?label1=?label2     && STRLEN(?label1)>3 )  ||
                 (?city1=?city2       && bound(?city1)     )  ||
                 (?country1=?country2 && bound(?country1)  )      )
}

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%    DBM-HAR-02
PREFIX dbo: <http://dbpedia.org/ontology/>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
SELECT  DISTINCT ?person ?provenance
WHERE {
      {#!reif(?id,?person,rdf:type,owl:Thing)!#
#!meta(?id,<http://ns.inria.fr/dbpediafr/voc#hasMainRevision>,?provenance)!#}
      #OPTIONAL
      {#!data(?person,?p,?place)!# #!data(?place,rdf:type,dbo:Place)!# }
      OPTIONAL
      {#!data(?place,owl:sameAs,?place2)!#}

      FILTER (    (bound(?place2) &&      EXISTS{#!data(?place2,dbo:isPartOf,<|region|>)!#} )
                   ||                           EXISTS{#!data(?place,dbo:isPartOf,<|region|>)!#}      )
}
```