Projet Web Sémantique

Les étudiants

Corentin Marionneau
Merlin Barzilai

Sont fiers de vous présenter
leur compte rendu de projet

Table des matières

Projet de Web sémantique

Introduction

Ce document présente l'évolution du projet de Web sémantique. Ce dernier avait pour objet la sémantisation de données depuis un dataset sous forme de fichier csv issu du site de données de l'ESR (Enseignement supérieur de Recherche) en données sémantiques 5 étoiles.

Notre projet est accessible dans son intégralité depuis le repository GitHub suivant : https://github.com/Slaanaroth/WebSemantic et la structure du repository est expliquée en détail dans le fichier README.md.

PS : Nous indiquerons entre parenthèses dans ce rapport l'emplacement dans le repository GitHub de tous les fichiers que nous évoquerons.

Présentation des données

Les données utilisée sont issues du site l'ESR. Il s'agit de données sur les groupes de recherche de l'enseignement supérieur bénéficiaires de la prime d'excellence scientifique en France entre 1993 et 2012. Le fichier était accessible au format CSV à l'adresse suivante : https://data.enseignementsup-recherche.gouv.fr/explore/dataset/fr-esr-pes-pedr-beneficiaires/

Ce fichier CSV comporte un total de 19 colonnes, chaque ligne représentant une prime d'excellence versée. Les informations disponibles sont notamment : l'année de la prime, le sexe des bénéficiaires, leur nombre, le domaine de recherche associé au projet, le corps de métier des bénéficiaires, l'établissement auquel ils appartiennent, l'académie dont dépend l'établissement, la région de l'établissement, le PRES (Pôle de Recherche d'Enseignement Supérieur) associé, la potentielle IDEX (Initiative D'EXcellence) ayant remis la prime ainsi que les coordonnées géographiques de l'établissement.

Transformation des données

Avant de pouvoir utiliser les données il a fallu les faire passer de leur format d'origine – CSV – au format de données sémantiques RDF. À cette fin nous avons utilisé l'outil TARQL. Cet outil permet d'utiliser les champs de chaque entrée d'un fichier CSV dans un script SPARQL afin de construire des triples formant un graphe RDF. Nous avons donc écrit un script SPARQL (building/builder.sparql dans le repository) permettant de créer des triples RDF depuis le fichier CSV.

```
 tarql --dedup 999999 -e utf-8 -d ";" builder.sparql fr-esr-pes-pedr-beneficiaires.csv > ../rdf/graph.ttl
```

Commande utilisée pour générer le graphe

Dans ce script SPARQL nous créons donc pour chaque ligne du CSV :

- un blank node représentant une prime (frapo:Funding)
- un blank node représentant un groupe de recherche (frapo:ResearchGroup)
- une ressource représentant l'établissement (dbo:EducationalInstitution)
- une ressource représentant le secteur de recherche (pext:ResearchArea)

- une ressource représentant le type de profession (pext:Profession)
- une ressource représentant l'IDEX (frapo:FundingAgency)
- une ressource représentant l'académie (dbo:EducationalInstitution)
- une ressource représentant le PRES (frapo:ResearchInstitute)
- une ressource représentant le région (dbo:Region)

Tous les vocabulaires de notre graphes (classes comme propriétés) ont été repris de vocabulaires existants afin d'améliorer la compatibilité entre notre graphe et les linked data. Nous avons principalement utilisé frapo (http://purl.org/cerif/frapo/), une ontologie fournissant du vocabulaire pour les projets de recherche ainsi que leur financement, mais également foaf pour les nom des différentes ressources et dbo (ontologie dbpedia) pour des termes génériques, notamment sur la région ou les institutions. Nous avons également utilisé quelques autres vocabulaires plus spécifiques tels que georss ou protonext.

Le graphe rdf en format Turtle résultant de cette conversion peut être trouvé sur notre repository dans le fichier rdf/graph.ttl

Cette première étape nécessitait également la création de 2 requêtes SPARQL intéressantes sur notre dataset, que nous avons écrites et testées (grâce à un endpoint sur un serveur local Fuseki), elles se trouvent dans le dossier queries/ de notre repository et seront détaillées (ainsi que leur résultat) plus tard dans ce rapport.

Liaison des données

Une fois notre graphe RDF généré, nous devions tenter de lier nos donner avec celles d'un second groupe afin d'effectuer une requête SPARQL mêlant plusieurs tables. Le groupe avec lequel nous avons décidé de collaborer est celui de Mica MENARD, Robin WIBAUX et Quentin LE GOUVELLO, détaillant les différentes initiatives contre le sexisme et le harcèlement sexuel dans différents établissement supérieur de France. Cette liaison de données nous semblait pertinente car notre dataset possédant des informations sur le genre des personnes bénéficiaires de primes d'excellence, il aurait été possible d'effectuer des études statistiques étudiant la répartition homme-femme en pourcentage des primes d'excellence reçues par sexe dans chaque établissement et la croisant avec les établissement possédant des initiatives anti-sexisme afin d'étudier s'il y a ou non une corrélation, on pourrait également voir l'évolution du ratio homme-femme bénéficiaires d'un établissement avant et après l'apparition d'une initiative anti-sexisme.

Nous avons donc récupéré leur graphe RDF (rdf/their_graph.ttl sur le repository) et avons tenté d'utiliser Jena-Fuseki afin d'installer sur un serveur local un endpoint SPARQL permettant d'effectuer des requêtes sur les deux datasets. Après plusieurs échec nous nous sommes tourné vers les query SPARQL dans Java grâce à Jena et ARQ. Dans un programme java (queries/MultigraphQuery.java) nous avons donc créé deux objets Model depuis les fichiers Turtle des deux graphes et les avons incorporés dans un Dataset en tant que graphes nommés. Nous avons ensuite utilisé ARQ pour effectuer la requête SPARQL trouvable en tant que queries/multigraph_query.sparql sur le repository. Cette requête a pour but de donner pour chaque établissement dans lequel une initiative anti-sexisme a lieu, le total d'hommes et de femmes ayant reçu des primes d'excellence, afin d'éventuellement pouvoir y observer une potentielle corrélation.

Nous avons donc obtenu le résultat suivant :

Institution	Gender	Beneficiaires
"Institut national des sciences appliquées de Lyon"	"Femmes"	35
"Institut national des sciences appliquées de Lyon"	"Hommes"	45
"Institut national des sciences appliquées de Rennes"	"Femmes"	13
"Institut national des sciences appliquées de Rennes"	"Hommes"	37
"Institut national des sciences appliquées de Rouen"	"Femmes"	4
"Institut national des sciences appliquées de Rouen"	"Hommes"	37
"Université Lille 1 - Sciences technologies"	"Femmes"	53
"Université Lille 1 - Sciences technologies"	"Hommes"	78
"Université Lille 3 - Charles-de-Gaulle"	"Femmes"	27
"Université Lille 3 - Charles-de-Gaulle"	"Hommes"	41
"Université d'Artois"	"Femmes"	19
"Université d'Artois"	"Hommes"	57
"Université d'Orléans"	"Femmes"	51
"Université d'Orléans"	"Hommes"	80
"Université de Bordeaux"	"Femmes"	4
"Université de Bordeaux"	"Hommes"	10
"Université de Lorraine"	"Femmes"	87
"Université de Lorraine"	"Hommes"	119
"Université de Poitiers"	"Femmes"	74
"Université de Poitiers"	"Hommes"	112
"Université de Rouen"	"Femmes"	66
"Université de Rouen"	"Hommes"	100
"Université de Strasbourg"	"Femmes"	97
"Université de Strasbourg"	"Hommes"	135
"Université de la Nouvelle-Calédonie"	"Hommes"	12
"Université de technologie de Belfort-Montbéliard"	"Femmes"	12
"Université de technologie de Belfort-Montbéliard"	"Hommes"	35
"Université du Havre"	"Femmes"	12
"Université du Havre"	"Hommes"	55
"École nationale supérieure d'ingénieurs de Caen"	"Femmes"	6
"École nationale supérieure d'ingénieurs de Caen"	"Hommes"	31
"École nationale supérieure de chimie de Rennes"	"Femmes"	5
"École nationale supérieure de chimie de Rennes"	"Hommes"	28
"École nationale supérieure de mécanique et d'aérotechnique de Poitiers"	"Femmes"	9
"École nationale supérieure de mécanique et d'aérotechnique de Poitiers"	"Hommes"	31
"École normale supérieure de Cachan"	"Femmes"	18
"École normale supérieure de Cachan"	"Hommes"	54
"École normale supérieure de Lyon"	"Femmes"	33
"École normale supérieure de Lyon"	"Hommes"	58

On peut y observer qu'aucun des établissement n'a plus de femmes bénéficiaires que d'hommes, ce qui, croisé avec d'autres statistiques, pourrait permettre de conclure si ces initiatives ont ou non une influence sur l'évolution de la proportion de femmes dans la recherche scientifique.

Ontologies et Inférences

Par la suite, une fois notre graphe rdf créé et testé, il nous fallait l'améliorer afin d'obtenir des données 5 étoiles, pour cela, nous devions tout d'abord écrire un schéma d'ontologie que nous pourrons par la suite utiliser afin de générer de nouveaux triples par inférence. Pour cela nous avons écrit notre schéma en Turtle dans le fichier inferences/ontology.ttl et y définissons que toutes les 8 classes que nous utilisons dans notre graphe sont des classes (?notreClasse rdf:type owl:Class) et pour chacune d'entre elle nous en définissions la classe mère grâce à rdfs:subClassOf. Pour chacune des propriétés utilisées dans notre graphe nous définissions qu'ils était de type rdf:Property et lui assignons son domaine dans notre graphe et parfois une range (lorsqu'il ne porte pas vers des littéraux), par exemple pour la propriété frapo:funds nous avons :

```
frapo:funds rdf:type rdf:Property ;
            rdfs:domain frapo:Funding ;
            rdfs:range frapo:ResearchGroup.
```

Une fois notre schéma écrit, nous avons créé une programme Java utilisant Jena (inferences/InferenceGeneration.java) dans lequel nous importons les deux graphes dans chacun un objet Model depuis leur fichier, puis nous générons un modèle d'inférence RDFS de type InfModel en utilisant la fonction createRDFSModel(schema, data). Nous ajoutons ensuite le modèle d'inférence au graphe initial et l'exportons dans un nouveau fichier Turtle : rdf/graphWithRDFS.ttl

Linked data

Notre graphe maintenant un peu plus détaillé grâce aux nouveaux triples RDF générés par inférence, nous avons donc ensuite abordé la dernière étape nécessaire à l'obtention de la 5e étoile des Linked Data : lier nos données au cloud de Linked Data déjà existant, pour ce faire nous avons voulu ajouter des propriétés owl:sameAs à nos ressources de type dbo:Region afin des les lier en tant qu'équivalents des ressources des régions françaises de dbpedia. Pour ce faire nous avons modifié notre script de construction nous ayant servi avec TARQL dans la première étape (nouveau fichier : building/builderSameAs.sparql) afin d'y incorporer pour chaque région un owl:sameAs. Pour cela nous avons ajouté à notre CONSTRUCT la ligne suivante :

```
  ?URI_Region owl:sameAs ?sameAsRegion.
```

Le ?URI_Region correspondant à l'URI de notre ressource dans notre graphe et ?sameAsRegion correspondant à l'URI de la région sur dbpedia, mappée dans le WHERE du script tel que :

```
    BIND (REPLACE(STR(?region),"^TOM .*","Overseas_territory_(France)")
as ?region_fixed)
    BIND (URI(CONCAT("dbfr:", ?region_fixed)) as ?sameAsRegion)
```

On peut voir qu'on remplace le nom de la région par Overseas_territory_(France) lorsque l'entrée de la région dans le CSV est « TOM + Collectivités territoriales » car il s'agît du seul nom de région ne correspondant pas à sa ressource dbpedia, et ensuite on forme l'URI tel que dbfr:NomDeLaRegion (dbfr étant le préfixe de http://fr.dbpedia.org/resource/).

Une fois ce script modifié, il suffisait de relancer la commande de TARQL (via building/buildSameAs.sh) en prenant comme script de construction le nouveau script SPARQL et comme destination un nouveau fichier Turtle : rdf/graphSameAs.ttl dans lequel nous avons ensuite généré à nouveau les inférences RDFS ce qui nous donne alors le fichier suivant : rdf/graphSameAsWithRDFS.ttl. Ce dernier contient donc un graph RDF de données 5 étoiles car il utilise un format standard, ouvert et sémantique (RDF Turtle) et est maintenant lié au cloud de Linked Data par dbpedia.

Métadonnées VOID

La dernière étape de ce projet étaient enfin d'ajouter à notre graphe des métadonnées décrivant notre dataset grâce au vocabulaire VOID (http://rdfs.org/ns/void#). Pour cela nous avons ajouté avec Jena (mais une requête SPARQL insert data aurait tout autant fait l'affaire) un certain nombre de triples décrivant notre Dataset. Le fichier java procédant à l'ajout peut être trouvé sur le repository dans void/MetadataGeneration.java et le résultat est dans rdf/graphFinal.ttl. L'équivalent de l'ajout via un insert data SPARQL est trouvable dans void/metadata.sparql.

Les triples RDF suivants ont donc été ajoutés :

```
  _:WebSemDataset a void:Dataset;
     void:feature <http://www.w3.org/ns/formats/Turtle>;
     void:triples ?nbTriples;
     void:classes ?nbClasses;
     void:properties  ?nbProps;
     dcterms:source  <https://data.enseignementsup-
recherche.gouv.fr/explore/dataset/fr-esr-pes-pedr-beneficiaires/>;
     dc:title "Bénéficiaires de la prime d'excellence scientifique";
     dc:description "Graphe rdf en format Turtle généré depuis le
dataset csv sur les primes d'excellence scientifique fourni par l'ESR";
     dc:creator _:MerlinBarzilai;
     dc:creator _:CorentinMarionneau.

  _:MerlinBarzilai a foaf:Person;
     foaf:name "Merlin Barzilai";
     foaf:mbox "merlin.barzilai@etu.univ-nantes.fr".

  _:CorentinMarionneau a foaf:Person;
     foaf:name "Corentin Marionneau";
     foaf:mbox "corentin.marionneau@etu.univ-nantes.fr".
```

Requêtes intéressantes

Voici donc les deux requêtes SPARQL que nous avons donc construites lors de l'étape 2 :

Requête 1

Cette première requête sélectionne le nombre de bénéficiaires de prime d'excellence scientifique chaque année par sexe, ce qui nous permet de voir chaque année la répartition homme-femme des bénéficiaires et l'évolution de cette répartition.

```
  PREFIX frapo: <http://purl.org/cerif/frapo/>
  PREFIX foaf: <http://xmlns.com/foaf/0.1/>
  PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

  SELECT (STR(?y) as ?Annee) (?g as ?Gender) (STR(count(?nb)) as ?
Beneficiaires)
  WHERE {
    ?prime frapo:hasAwardDate ?y.
    ?prime frapo:funds ?group.
    ?group foaf:gender ?g.
    ?group frapo:hasHeadcount ?nb.
  }
  group by ?y ?g
  order by ?y ?g
```

queries/query1.sparql

Cette requête a le résultat suivant :

Annee	Gender	Beneficiaires
"1993"	"Femmes"	"129"
"1993"	"Hommes"	"264"
"1994"	"Femmes"	"227"
"1994"	"Hommes"	"401"
"1995"	"Femmes"	"74"
"1995"	"Hommes"	"191"
"1996"	"Femmes"	"130"
"1996"	"Hommes"	"270"
"1997"	"Femmes"	"167"
"1997"	"Hommes"	"324"
"1998"	"Femmes"	"186"
"1998"	"Hommes"	"343"
"1999"	"Femmes"	"207"
"1999"	"Hommes"	"371"
"2000"	"Femmes"	"225"
"2000"	"Hommes"	"356"
"2001"	"Femmes"	"198"
"2001"	"Hommes"	"357"
"2002"	"Femmes"	"210"
"2002"	"Hommes"	"354"
"2003"	"Femmes"	"207"
"2003"	"Hommes"	"369"
"2004"	"Femmes"	"208"
"2004"	"Hommes"	"337"
"2005"	"Femmes"	"220"
"2005"	"Hommes"	"361"
"2006"	"Femmes"	"250"
"2006"	"Hommes"	"414"
"2007"	"Femmes"	"251"
"2007"	"Hommes"	"389"
"2008"	"Femmes"	"272"
"2008"	"Hommes"	"416"
"2009"	"Femmes"	"267"
"2009"	"Hommes"	"434"
"2010"	"Femmes"	"261"
"2010"	"Hommes"	"428"
"2011"	"Femmes"	"277"
"2011"	"Hommes"	"423"
"2012"	"Femmes"	"286"
"2012"	"Hommes"	"420"

Requête 2

Cette seconde requête sélectionne pour chaque année le nombre de primes distribuées pour chaque domaine de recherche

```
PREFIX frapo: <http://purl.org/cerif/frapo/>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
```

```
  select (STR(?y) as ?Year) (?areaName as ?Area) (STR(count(?p)) as ?
Primes)
  where {
    ?prime frapo:hasAwardDate ?y.
    ?prime frapo:funds ?group.
    ?group frapo:hasExpertise ?ar.
    ?ar foaf:name ?areaName.
  }
  group by ?y ?areaName
  order by ?y ?areaName
```

queries/query2.sparql

Cette requête a (en partie) le résultat suivant :

Year	Area	Primes
"1993"	"Droit et sciences économiques"	57
"1993"	"Lettres et sciences humaines"	98
"1993"	"STAPS"	10
"1993"	"Santé"	36
"1993"	"Sciences"	192
"1994"	"Droit et sciences économiques"	85
"1994"	"Lettres et sciences humaines"	137
"1994"	"STAPS"	24
"1994"	"Santé"	69
"1994"	"Sciences"	313
"1995"	"Droit et sciences économiques"	30
"1995"	"Lettres et sciences humaines"	66
"1995"	"STAPS"	4
"1995"	"Santé"	15
"1995"	"Sciences"	150
"1996"	"Droit et sciences économiques"	39
"1996"	"Lettres et sciences humaines"	96
"1996"	"STAPS"	5
"1996"	"Santé"	28
"1996"	"Sciences"	232
"1997"	"Droit et sciences économiques"	68
"1997"	"Lettres et sciences humaines"	108
"1997"	"STAPS"	13
"1997"	"Santé"	49
"1997"	"Sciences"	253
"1998"	"Droit et sciences économiques"	61
"1998"	"Lettres et sciences humaines"	109
"1998"	"STAPS"	27
"1998"	"Santé"	56
"1998"	"Sciences"	276
"1999"	"Droit et sciences économiques"	75
"1999"	"Lettres et sciences humaines"	132
"1999"	"STAPS"	16
"1999"	"Santé"	54
"1999"	"Sciences"	301
"2000"	"Droit et sciences économiques"	71
"2000"	"Lettres et sciences humaines"	138
"2000"	"STAPS"	19
"2000"	"Santé"	49
"2000"	"Sciences"	304

[…]

Conclusion

Pour conclure, on peut dire que ce projet nous a permis de faire le tour du sujet à propos des données sémantiques 5 étoiles, du passage de CSV à données sémantique en RDF, l'ajout de triples par inférence depuis un schéma, l'ajout de métadonnées décrivant le dataset et l'incorporation des données au sein du cloud de Linked Data. Nous avons également appris à manipuler les données RDF grâce à SPARQL ainsi que faire des requêtes sur plusieurs graphes à la fois.

Comme piste d'amélioration à notre projet, on pourrait éventuellement envisager l'ajout de plus de liaison au cloud, avec plus de owl:sameAs sur d'autres ressources, les établissements par exemple.