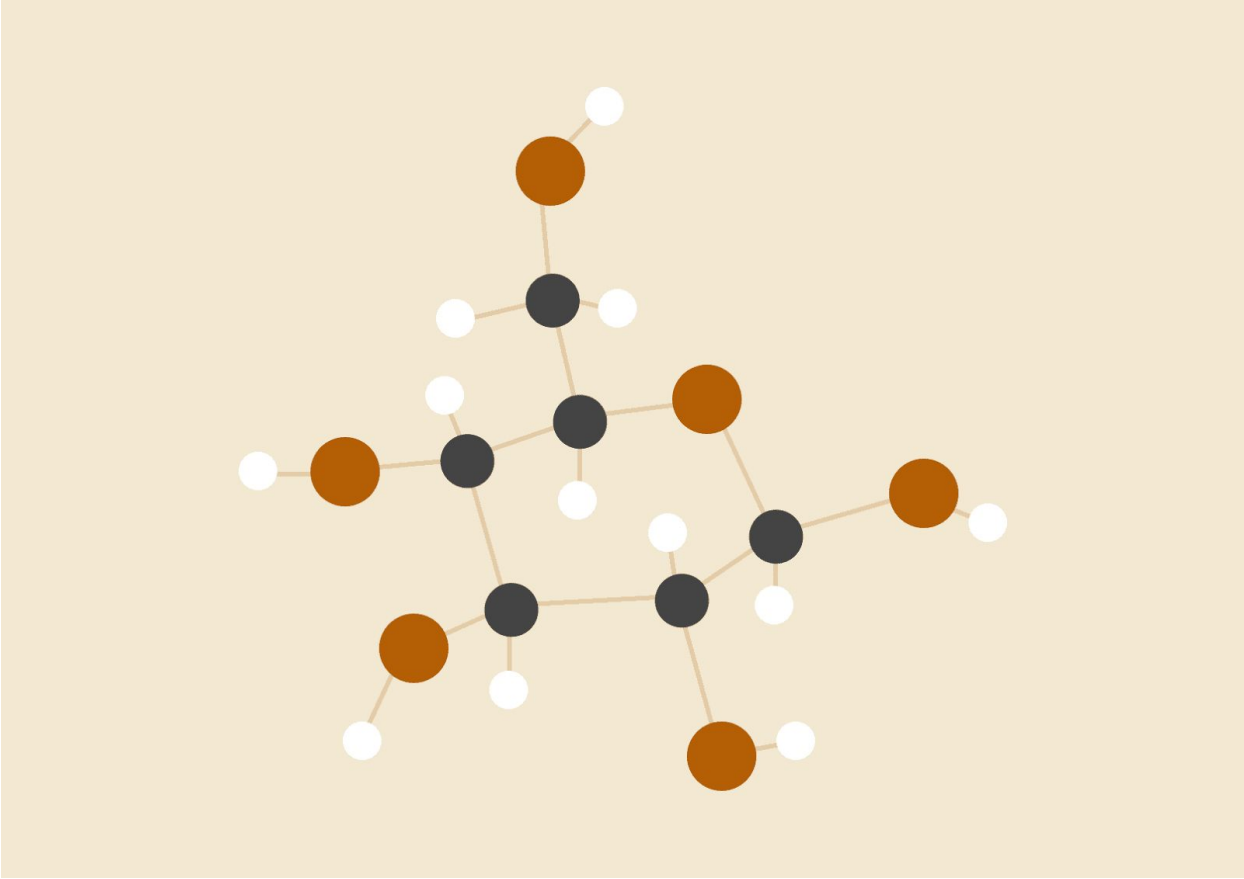
RAPPORT PROJET WEB SÉMANTIQUE

Des données ouvertes vers des données en 5 étoiles



Maëlle Brassier, Florian Lardy, Ophélie Marinier

08/12/2017

WEB SÉMANTIQUE – M1 ALMA

INTRODUCTION

L’objectif de ce projet est de transformer les données ouvertes de l'Enseignement supérieur, de la Recherche et de l'Innovation en données sémantiques et de lier ses données sémantiques au cloud de “Linked Data : Connect Distributed Data across the Web”.

Le résultat final de ce projet, comme son nom l’indique, est d’établir un processus valide et efficace qui permettrait de rendre ces données ouvertes en des données 5 étoiles. Rappelons ce que sont des données 5 étoiles : ce sont des données publiées sur le Web, structurées et dans un format ouvert et non-propriétaire. De plus, il est nécessaire d’utiliser des URI afin que des tierces personnes puissent faire des références et enfin, les données doivent être liées à d’autres données.

Ce rapport présente les différentes étapes ainsi que nos choix de conception, démarches, observations et problèmes que nous avons pu rencontrer.

ETAPE 1

Choix de notre dataset

Pour ce projet nous avons choisi un dataset qui représente la liste des nouveaux membres actifs de l'Institut Universitaire de France (IUF) par année d'intégration (https://data.enseignementsup-recherche.gouv.fr/explore/dataset/fr-esr-iuf-les-membres/information/).

Notre dataset contient au total 2041 enregistrements, correspondant à chaque membre inscrit. Il comporte des informations sur les nouveaux membres de l’Institut Universitaire dans toute la France (tels que leurs noms, date d’arrivée, catégorie…), sur les établissements auxquels ils sont affectés (régions, coordonnées…) et les disciplines de chacun d’eux (secteur disciplinaire, groupe cnu…).

Afin d’obtenir un dataset optimal, il nous a semblé important d’effectuer un nettoyage préalable des données. De ce fait, nous avons enlevé tous les attributs id qui ne nous semblaient pas pertinents, à savoir tous les id ; puis nous avons fait quelques changements mineurs tel que des changements de caractères spéciaux. Il est à noter

également que l'attribut d'origine Géo-localisation comportait la latitude et la longitude, que nous avons préféré séparés en deux attributs distincts.

Sémantisation de nos données

Pour transformer nos données structurées en RDF nous avons utilisé `Tarql`. Pour ce faire, nous avons créé un fichier *.sparql* (*construct.sparql*) dans lequel nous utilisons un CONSTRUCT. Ce dernier est constitué de 3 BIND :

```
BIND (URI(CONCAT('http://iuf.org/Member/', ?nom, '_', ?prenom)) AS
?URIMember)
BIND (URI(CONCAT('http://iuf.org/Etablissement/', ?etablissement)) AS
?URIEtab)
BIND (URI(CONCAT('http://iuf.org/Discipline/',
?discipline_nomenclature)) AS ?URIDiscipline)
```

qui correspondent aux 3 sous parties que nous souhaitons obtenir, à savoir les informations sur les membres, sur les établissements et enfin les disciplines.

Il nous a semblé intéressant de lier ces sujets par des relations. Ainsi, un membre fait partie d'un établissement via le prédicat isInOrganisation et dans une discipline via inDiscipline.

Pour la construction de nos données en RDF nous avons utilisé du vocabulaire partagé comme *foaf, dbo, rdf* et *geo*, mais nous avons également créé notre propre vocabulaire appelé iuf comme ci-dessous.

```
@prefix dbo:  <http://dbpedia.org/ontology/> .
@prefix geo:  <https://www.w3.org/2003/01/geo/> .
@prefix iuf:  <http://ex.org/iuf/> .
@prefix rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix foaf:  <http://xmlns.com/foaf/0.1/> .
```

- Le vocabulaire dbo a été utilisé pour les propriétés : *year, region* et *educationalInstitution*.
- Le vocabulaire geo a été utilisé pour les données géographiques telles que *lat* et *long*.
- Le vocabulaire iuf est, comme précisé ci-dessus, notre propre vocabulaire. Il

comporte donc tous les attributs pour lesquels nous n'avons pas trouvé d'équivalent.
- Et enfin le vocabulaire foaf englobe toutes les informations relatives à une personne comme *lastName et givenName*.

Le fichier que nous avons obtenu via Tarql était sous forme Turtle mais nous avons décidé également de le convertir en format RDF/XML en utilisant RDF Converter afin de pouvoir exécuter nos requêtes via ARQ.

Nous avons donc deux fichiers : c*onstructTurtle.ttl* (que nous utiliserons pour définir nos ontologies) et *constructRDFXML.rdf* (pour les requêtes)

Les requêtes sur notre dataset

Nous avons conçu 3 requêtes différentes, en essayant d'utiliser le plus d'opérations disponibles du langage SPARQL (COUNT, OPTIONAL, FILTER..) . Comme mentionné précédemment, nos requêtes ont été faites via ARQ de Jena mais également Fuseki une fois que nous avons pu stocker nos graphes dans le serveur Fuseki.

1ère requête : La 1ère requête renvoie le top 10 des établissements où il y a le plus de femmes. (Pour voir son contenu : requête1.qr)

etablissement	countGlobal	countHomme	countFemme
"Universite Paris Diderot"	102	78	24
"Universite de Grenoble Alpes"	105	82	23
"Universite Paris 1 - Pantheon Sorbonne"	67	45	22
"Universite Pierre et Marie Curie"	105	86	19
"Universite Paris Ouest Nanterre La Defense"	49	30	19
"Universite Lille 3 - Charles-de-Gaulle"	28	11	17
"Universite Paris-Sud"	78	61	17
"Aix-Marseille universite"	85	68	17
"Universite Paris-Sorbonne"	43	29	14
"Universite Claude Bernard - Lyon 1"	44	30	14

2ème requête : La 2ème requête renvoie le pourcentage de senior par année. (requête2.qr)

annee	countotal	countsenior	countjunior	pourcentage_senior
"1991"	61	29	32	47.54098360655737704918032
"1992"	64	39	25	60.9375
"1993"	70	34	36	48.57142857142857142857142
"1994"	70	39	31	55.71428571428571428571428
"1995"	58	31	27	53.44827586206896551724137
"1996"	70	39	31	55.71428571428571428571428
"1997"	74	46	28	62.16216216216216216216216
"1998"	70	39	31	55.71428571428571428571428
"1999"	73	43	30	58.90410958904109589041095
"2000"	77	47	30	61.03896103896103896103896
"2001"	81	44	37	54.32098765432098765432098
"2002"	77	43	34	55.84415584415584415584415
"2003"	65	35	30	53.84615384615384615384615
"2004"	58	30	28	51.72413793103448275862068
"2005"	76	39	37	51.31578947368421052631578
"2006"	126	53	73	42.06349206349206349206349
"2007"	120	50	70	41.66666666666666666666666
"2008"	124	57	67	45.96774193548387096774193
"2009"	163	66	97	40.49079754601226993865030
"2010"	188	78	110	41.48936170212765957446808
"2011"	189	78	111	41.26984126984126984126984
"2012"	200	82	118	41.0
"2013"	134	48	86	35.82089552238805970149253
"2014"	131	47	84	35.87786259541984732824427
"2015"	115	41	74	35.65217391304347826086956
"2016"	134	48	86	35.82089552238805970149253

3ème requête : (Celle ci a été faite sous Fuseki) Renvoie le nom (concaténation de prénom + nom) des membres faisant partie du groupe cnu “Physique” par ordre d’année et leur groupe de corps s’ils en ont un. (requête3.rq)

	name	cnu	annee	groupe_corps
1	"DAVIER MICHEL"	"Physique"	"1991"	
2	"GALLET FRANCOIS"	"Physique"	"1991"	
3	"HAROCHE SERGE"	"Physique"	"1991"	
4	"LABASTIE PIERRE"	"Physique"	"1991"	
5	"BREZIN EDOUARD"	"Physique"	"1991"	"Professeurs et assimiles"
6	"GABAY MARC"	"Physique"	"1991"	"Professeurs et assimiles"
7	"JOANNY JEAN-FRANCOIS"	"Physique"	"1991"	"Professeurs et assimiles"
8	"SALATI PIERRE"	"Physique"	"1991"	"Professeurs et assimiles"
9	"COUDER YVES"	"Physique"	"1992"	
10	"HANSEN JEAN-PIERRE"	"Physique"	"1992"	
11	"BLAVETTE DIDIER"	"Physique"	"1992"	"Professeurs et assimiles"
12	"CASTAING BERNARD"	"Physique"	"1992"	"Professeurs et assimiles"
13	"FAUVE STEPHAN"	"Physique"	"1992"	"Professeurs et assimiles"

ETAPE 2

L'autre groupe avec lequel nous avons lié nos données est celui de Félix Jamet et Mattis Le Falhun. En effet, leur dataset porte sur les Enseignants titulaires de l'enseignement supérieur public (national), ce qui est très proche de nos données.

Afin de lier nos données, nous avons récupéré leur dataset que nous avons stocké dans notre server Fuseki à l'adresse localhost:3030/data_mattis_felix. Il suffit alors de modifier notre ancien fichier de CONSTRUCT en rajoutant des owl:sameAs pour chaque attributs communs.

Afin de créer un lien vers leurs données, nous voulions utiliser des owl:sameAs directement sur les URI. Malheureusement, nous nous sommes pas mis d'accord sur ce que l'on devrait utiliser afin de créer nos URI respective ce qui fait que nous n'avions plus la possibilité retrouver facilement les URI des données communes. De plus, notre groupe avait choisi de ne pas garder les codes / id des différents éléments car nous ne jugions pas cela pertinent dans notre dataset. Cela fut une erreur nous bloquant encore plus sur la liaison des datasets.

Si nous avions réussi à lier les données, les requêtes que nous aurions pu effectuer aurait eu besoin d'un SERVICE<http:localhost:3030/data_mattis_felix>

Une idée de requête aurait été de comparer le taux de femmes présentes dans une académie pour les membres de l'IUF aux enseignants titulaires de l'enseignement supérieur public.

ETAPE 3

Nous avons vite remarqué que nous avions besoin de créer notre ontologie afin d'obtenir de meilleurs résultats.

La création de notre ontologie se fait via un fichier model.ttl (model.ttl), dans lequel nous définissons notre vocabulaire. Comme notre CONSTRUCT, il se décompose plus ou moins en 3 parties, pour les membres, disciplines et organisations.

iuf:Membre, iuf:Discipline et iuf:Organisation sont considérés comme des classes puisqu'ils représentent des sujets. Ils ont également des équivalences avec des vocabulaires déjà existants, respectivement foaf:Person, vaem:Discipline et dbo:Organisation.

Nos prédicats sont considérés comme des DatatypeProperty qui ont chacun un domaine et un range ; le domaine étant le sujet qui lui correspond, et le range l'objet (très souvent un Literal).

Enfin, les prédicats que nous avons créé permettant de lier un membre à un établissement une discipline sont des ObjectProperty.

Notre fichier contenant nos règles d'inférence se nomme regles.rules. Il permet d'ajouter des triplets lorsque certains coïncident avec eux. Nous y avons ajouté des règles de base et nous avons également ajouté des règles liées à notre propre vocabulaire. Voici quelques exemples de règles :

Premier exemple : si region est une sous classe de PopulatedPlace et que PopulatedPlace est une sous classe de place, alors region est une sous classe de place. Cette règle se traduit comme cela :

[(dbo:region rdfs:subClassOf dbo:PopulatedPlace) (dbo:PopulatedPlace rdfs:subClassOf dbo:place) -> (dbo:region rdfs:subClassOf dbo:place)]

Deuxième exemple : si section_cnu est une sous propriété de groupe_cnu et que

groupe_cnu est une sous propriété de secteur_disciplinaire, alors EducationalInstitution est une sous classe de Agent. Cette règle se traduit comme cela :

[(iuf:section_cnu rdfs:subPropertyOf iuf:groupe_cnu) (iuf:groupe_cnu rdfs:subPropertyOf iuf:secteur_disciplinaire)-> (iuf:section_cnu rdfs:subPropertyOf iuf:secteur_disciplinaire)]

Pour créer des inférences dans notre dataset, nous avons également utilisé l'outil Fuseki qui possède un moteur d'inférence. On peut donc lui passer en paramètre notre ontologie et nos règles pour qu'il puisse en créer sur notre dataset.

Ainsi nous avons obtenu un dataset plus riche qu'au départ grâce aux inférences faites par Fuseki. En effet nous pouvons le voir grâce au nombre de lignes affichées sur Fuseki.

ETAPE 4

Afin d'avoir des données acceptable et de les lier au cloud de linked data, il faut suivre certaines règles :

-Utiliser des URI afin d'identifier les données

-Utiliser des URI HTTP afin de permettre un accès facile à tous

-Fournir des informations RDF utiles à toutes personnes regardant un des URI

-Inclure des liaisons vers d'autres URI

Nous avons "créé" nos propre URI HTTP, il faudrait en plus avoir un domaine afin de les enregistrer en ligne.

Comme nous avons utilisé des ontologies de DBpedia, il serait également possible de lier nos données au cloud de linked data via nos années, nos régions, nos institutions.

ETAPE 5

Pour la description du dataset à l'aide du vocabulaire VoID, nous avons utilisé les préfixes suivants : dcterms, foaf, rdfs, owl et donc void. Cela nous a permis de définir un titre, une description, une homepage, un contributeur, des éditeurs (publisher), une

licence ainsi que le nombre de triple.

Ensuite, nous avons défini plus en détails le contributeur ainsi que les éditeurs.

Enfin, nous avons défini le lien vers le dataset du groupe de Mattis et Félix.

CONCLUSION

Malgré le fait que nous avions vu la théorie en cours, nous avons eu du mal à l'appliquer à travers ce projet. Ce qui nous a fait perdre un temps certain car nous n'avions trouvé que peu d'informations sur internet.

Cependant ce projet était très intéressant car nous avons vu comment il était possible d'améliorer des données du web actuel en les modifiant pour qu'elles passent de données 3 étoiles en données 5 étoiles. En effet, au début nous avions des données déjà en 3 étoiles.

Pour rappel, la première étoile permet de savoir si des données publiées sur le web sont sous licence ouverte. Comme nous avons récupéré nos données sur le site Open Data du gouvernement, alors nous avions déjà cette étoile. Pour avoir la deuxième étoile il fallait que nos données soient structurées et c'était bien notre cas car qu'elles étaient au format CSV. De plus, le format CSV est non propriétaire donc nous avions également la troisième étoile.

Pour obtenir les deux dernières étoiles, nous avons dû faire des changements. La quatrième étoile impose de suivre les standards RDF et d'utiliser des URIs ce qui est notre cas. Enfin, pour finir, nous avons liés nos données. Ainsi nous avons obtenu la cinquième étoile.

Nous pouvons donc conclure que nous avons transformé nos données en 5 étoiles.