COUILLEROT Carol
MAHIER Loïc
PHALAVANDISHVILI Demetre

Rapport de projet *

*rapport réalisé sous LaTeX

Sommaire

1. Introduction

L'objectif de ce projet est de transformer des données ouvertes de l'Enseignement supérieur, de la Recherche et de l'Innovation (https://data.esr.gouv.fr/FR/) en données sémantiques et de lier ses données sémantiques aux données sémantiques de nos collègues. Pour ce faire nous avons choisis des données sur les budgets dédiés à la recherche et au transfert de technologie (R&T) des collectivités territoriales.

2. Sémantisation des données

2.1 Nettoyage des données

Les données étant choisis, il est désormais temps de les sémantiser. Cependant, un peu de "nettoyage" s'impose avant cela, en effet le jeu de données présente certain défaut à corriger. Ainsi nous avons dû normaliser les années, les budgets et les régions. Par exemple il y avait plusieurs régions que l'on trouvait avec différentes syntaxes, cela aurait posé des problèmes plus tard au niveau des requêtes. Autre défaut du jeu de données, la présence d'une valeur "TOTAL FRANCE" dans les régions qui prête à confusion. Nous avons donc créé en plus de la colonne région une colonne pays servant à distinguer les deux échelles géographique. Enfin, nous avons créé une colonne pourcentage. En effet, il était auparavant spécifié entre parenthèse dans les champs "indicateur" et "objectif" si le budget en question est un montant (en euro) ou un pourcentage. Il est désormais possible de le savoir à l'aide d'un simple booléen dans la colonne pourcentage. Toutes ces modification sont apportées dans le but de rentre le jeu de données davantage utilisable et ainsi de simplifier nos futures requêtes.

2.2 Choix des ontologies

Voila les modifications majeures apportées au jeu de données. Maintenant que les données sont "propres" et pleinement exploitable on peut les séman-

tiser. La première étape est d'identifier les ontologies adéquates.

On commence par distinguer que nous avons des données qui renseignent sur la région, la collectivité, l'année, le code, le pourcentage, etc. Pour ces données là, l'ontologie "dbo" propre à http://wiki.dbpedia.org/ est intéressante : nous utilisons ainsi les prédicats "dbo" suivants : dbo :year, dbo :country, dbo :region, dbo :code, dbo :Organization et dbo :percentage. Ces champs sont en effet assez simple pour certain : une année, un code et un pourcentage par exemple sont des prédicats basiques présents dans dbpedia qui nous satisfont et qui correspondent aux données sans perdre de l'information. De même, pour les régions qui sont référencées dans dbpedia. A contrario le prédicat dbo :Organization que nous utilisons pour les collectivités territoriales, générales et régionales n'est peut être pas le plus adéquat car assez vague. Mais nous l'avons conservé faute d'avoir trouvé mieux.

Ensuite on remarque que le reste des données identifie un budget, sa valeur ou son pourcentage, son objectif et son application. Pour ce faire nous avons choisis d'utiliser l'ontologie "frapo" (http://www.sparontologies.net/ontologies/frapo/source.html qui sert notamment à définir des budgets. En effet nous aurions pu par exemple encore utiliser des ontologies de dbpedia : cependant celles-cis ne permettaient pas d'avoir des prédicats assez précis pour définir le type de nos données. Nous avons donc décidé d'utiliser "frapo". Ainsi nous avons besoin des prédicats suivants : frapo :BudgetInformation, frapo :appliesFor et frapo :BudgetedAmount.

2.3 Écriture du "construct"

Passons à présent à l'écriture du fichier "construct" qui va générer nos données RDF à partir du fichier CSV. Celui-ci se fait assez facilement, nous allons le voir en détail en procédant par étape. On va y trouver trois morceaux, un premier avec tous les préfixes utilisés nommé "PREFIX", un deuxième nommé "CONSTRUCT" et un dernier nommé "WHERE". Une fois n'est pas coutume, commençons par le dernier. La clause "WHERE" va servir à récupérer les valeurs des différentes colonnes du jeu de données et à les attribuer à des variables à l'aide du mot clé "BIND". Variables que nous typons à la manière du XMLSchema. On définit ainsi des xsd :integer et des sxd :string par exemple. Dans le même temps, on crée notre URI en ajoutant à une url que nous avons arbitrairement choisis autant de variable que nécessaire pour identifier chacun des tuples de notre jeu de données. L'objectif étant

de ne pas avoir de doublon. Pour ce faire nous utilisons une conditionnelle (IF(BOUND(x,y))) puisque nous avons besoin d'un champ qui est parfois vide. Cela nous permet selon la situation de passer d'un modèle d'URI à une autre suivant la conditionnelle.

Passons à la clause "CONSTRUCT". Celle-ci est finalement triviale, nous y appliquons simplement les prédicats définis un peu plus tôt à chacune des variables que nous avons créé dans la clause "WHERE". Et il n'y a pas grand chose de plus à en dire. Terminons donc l'explication du "construct" en présentant le premier morceau, à savoir celui qui énonce tous les préfixes. On y retrouve tous les liens vers les ontologies que l'on utilise : notamment vers dbo, frapo, rdf et xsd.

2.4 Obtention des données au format RDF

Maintenant notre "construct" écrit, nous créons nos données RDF à l'aide de l'applet TARQL, via le terminal. Nous indiquons l'encodage de nos données, le caractère qui sert à délimiter les données, le fichier "construct", le fichier au format CSV contenant les données et enfin le fichier de destination des données RDF obtenues à l'aide d'un flux de redirection :
tarql -e utf-8 –delimiter \; construct.sparql data_non_semantiques.csv > data_semantiques.rdf

2.5 Écriture de quelques requêtes

3. Liaisons des données

3.1 Présentation des jeux de données liés

Maintenant nos données propres, sémantisées et testées, il est temps de les lier à celles de nos camarades. Ainsi nous avons lié nos données à celles de deux autres groupes. Le premier groupe a des données qui portent sur les moyens consacrés à la R&D par les entreprises. De ce fait, lier nos données à

ce groupe nous permet d'avoir à l'échelle nationale à la fois les budget public (nous) et privé (eux). Des lors nous constatons qu'ils possèdent le code et le nom de chaque région, nous savons donc déjà sur quel attribut nous allons pouvoir lier nos données entre nous. C'est en gardant cette attribut en tête que nous avons cherché à lier nos données à un second groupe. Ce second groupe possède des données sur le nombre d'étudiants inscrits dans l'enseignement supérieur que nous avons évidemment lié à l'aide du code région. Ces données nous permettent de regardé si il y a un lien entre l'investissement dans la recherche et le nombre d'étudiant (étudiant potentiellement destiné à être chercheur) par région. Nous constatons que beaucoup de groupe ont des données possédant des codes régions ou simplement le noms de la région. Il serait donc possible de lier énormément de jeux de données entre eux. Nous reviendrons dessus un peu plus tard et expliquerons pourquoi nous n'avons pas cherché à lier davantage de données entres elles.

3.2 Modification du "construct"

3.3 Écriture d'une requête fédéré

4. Inférence

5. VOID

6. Conclusion

7. Annexe