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 commencer par normaliser les années 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 ce qui prête à confusion. Nous avons donc créé en plus de la colonne région une colonne pays servant à distinguer les deux échelles géographique.

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 sémantiser. 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 et l'année. 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.

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. Nous avons donc besoin des prédicats suivants : frapo :BudgetInformation, frapo :appliesFor et frapo :BudgetedAmount.

2.3 Écriture du "construct"

L'écriture du "construct" se fait assez facilement. Tout d'abord, on ajoute dans la clause "WHERE" des "BIND" qui vont nous servir à récupérer les valeurs dans chaque colonne du jeu de données. Nous attribuons ainsi une variable à ces valeurs. Variable que nous typons à la manière du XMLSchema. 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 avons utilisé une conditionnelle puisque nous avons besoins d'un champ qui est parfois vide. Ensuite nous passons dans la clause "CONSTRUCT" où nous allons appliquer le prédicat (choisis ci-dessus) à chacune de nos variables.

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

3. Liaisons des données

4. Inférence

5. Conclusion

6. Annexe