

ENSG
Géomatique

ÉCOLE NATIONALE
DES SCIENCES
GÉOGRAPHIQUES

Construction d'un extracteur d'information géographique Avec R et SPARQL

Juliette Delannoy

Projet de développement informatique de troisième année du cycle ingénieur

Encadré par Hadrien Commenges et Thomas Louail

Résumé

Depuis plusieurs années, le web sémantique a fait son apparition avec l'idée de structurer la donnée pour pouvoir automatiser son utilisation. Par exemple, le projet DBpedia a été créé en marge de Wikipédia pour stocker l'information brute disponible sur cette plateforme dans une base de données. Bien que libre, ces données sont difficiles d'accès car il faut s'être familiarisé au web sémantique, à DBpedia et au langage SPARQL permettant de faire des requêtes sur la base de données. L'objectif de mon stage est donc de fournir un outil simple d'utilisation avec une prise en main rapide, qui permette d'interroger la base de données sans être familier des contextes cités plus haut.

Détails du projet

Le projet a été mené par Juliette Delannoy, étudiante en 3^{ème} année d'école d'ingénieur à l'ENSG, filière Carthagéo, dans le cadre du projet développement.

Ce projet a été encadré par Hadrien Commenges et Thomas Louail, dans les locaux du laboratoire de recherche Géographie-cités, 13 rue du Four, 75006 Paris.

Ce projet a duré 5 semaines, du 8 janvier 2018 au 9 février 2018.

Plan

Résumé	1
Détails du projet	1
Plan	2
Introduction générale	4
Du web sémantique à DBpedia	5
Le Web Sémantique c'est quoi ?	5
Normaliser le format des données	5
Interroger une base de données	6
Vocabulaire standard	7
Où la donnée est-elle stockée ?	7
La base de données DBpedia	9
Organisation de l'interface	10
Guide utilisateur	10
Requête	11
Carte	11
Informations supplémentaires	13
Fonctionnement de l'interface	14
Sujet	14
Type principal	14
Type secondaire	16
Précision du nom	17
Prédicat et objet	18
Choix du prédicat	18
Prédicat nommé- choix de l'objet	21
Deuxième prédicat	22
Résultats	23
Nombre de résultats	23
Export	24
Spatialisation des résultats	24
Potentiels et limites	25
Problèmes liés à DBpedia	25
Choix d'extraction des ontologies	27

Choix des délimitations du projet	28
Conclusion	30
Sitographie	31
Annexes	32
Annexe 1 : Types des sujets.....	32
Annexe 2 : Liste des prédictats	32
Annexe 3 : Dictionnaire des prédictats.....	32
Annexe 4 : Résultats de requête au format PDF	32
Annexe 5 : GANTT.....	32

Introduction générale

Le développement en flèche d'internet ces dernières années et l'apparition de nouvelles sources de données massives (big data) permet à chacun d'avoir à disposition une quantité quasi-illimitée de données, la difficulté majeure étant de savoir où chercher l'information qui nous intéresse ainsi que comment. Depuis une dizaine d'années est apparu le concept du web sémantique, dont l'idée est de structurer au mieux les données pour permettre l'accès à la donnée brute. Cela rendrait possible l'interrogation automatique et pourrait notamment permettre aux machines de devenir plus autonomes en leur permettant de faire des recherches standardisées sur internet.

Une des bases de données structurées les plus connues est celle de DBpedia, qui contient une grande partie des informations disponibles sur Wikipédia. Afin de pouvoir utiliser cette base de données structurée du web, ainsi que d'autres, un langage permettant de les interroger est nécessaire. Le langage SPARQL permet d'interroger les bases de données, mais, comme tout langage, il nécessite un apprentissage préalable pour obtenir les résultats souhaités.

Mais quel est l'intérêt de DBpedia ? Lorsqu'on a besoin d'une information simple telle que les noms des albums de notre chanteur préféré, on peut les trouver très facilement sur Wikipédia. Mais quand on cherche quelque chose comme : « les athlètes nés en France depuis 1950 », cela devient plus compliqué. Il serait donc utile d'avoir un outil d'extraction d'information, potentiellement géographique, qui permette d'avoir une interface utilisable par n'importe qui et permettant, en quelques clics, d'avoir n'importe quel type d'information. Le projet mené pendant ces 5 semaines en était le but.

Pour construire cet extracteur d'information géographique, il a fallu se familiariser avec les concepts du web sémantique, apprendre le langage de requêtes SPARQL, délimiter et planifier le projet, puis mettre en place l'interface. Le temps nécessaire à chacune des étapes, ainsi que le détail de chacune d'entre elles sont présentés en annexe sous forme d'un diagramme de GANTT.

Je commencerai donc par définir le web sémantique et ses objectifs, puis je présenterai l'organisation de l'interface avant de détailler son fonctionnement et l'utilisation du langage SPARQL. Je finirai enfin par le potentiel et les limites de mon interface.

Du web sémantique à DBpedia

De plus en plus de données étant disponibles sur internet, il y a une réelle nécessité de traiter et d'interpréter ces données. Or ces données ne sont pas toujours structurées et donc difficilement lisibles par les ordinateurs. Un exemple de cela est celui des sites de comparaison des prix des vols selon les compagnies aériennes. Dans ces compagnies, un développeur est employé pour étudier la structure de chacun des sites de vols d'avion et ainsi de permettre l'extraction de son contenu. Avec un web uniformément structuré, ce travail ne serait pas nécessaire puisque l'architecture de toutes les pages serait similaire.

On s'oriente alors vers une transition d'un web dit « sémantique ».

Le Web Sémantique c'est quoi ?

Le web sémantique, ou **Web 3.0** est souvent appelé le web des données.

Le but est de pouvoir utiliser le web comme une immense base de données, que l'on peut enrichir (les projets gouvernementaux fournissent des jeux de données utilisables par tous), ou utiliser (recherche d'information pour enrichir sa propre base de données).

L'idée est de parvenir à un Web intelligent, où les pages des sites seraient gérées par une base de données intelligente ce qui permettrait de mettre les objets au service des personnes.

Pour cela il faut que les données soient interopérables et soient accompagnées de leur sémantique (**ontologie**). L'ontologie constitue un modèle de données qui représente les concepts d'un domaine ainsi que les relations entre ces concepts. Les ontologies permettent de représenter la donnée pour qu'elle soit compréhensible par un ordinateur. Il s'agit d'une sorte d'étiquette : Usain Bolt est une personne et un athlète, il s'agit là de deux ontologies.

En plus de données interopérables, il est nécessaire d'utiliser un dispositif langagier normalisé afin qu'il soit utilisable par tous de la même manière, ce qui permettra l'accès universel et l'utilisation intelligente de la donnée par les ordinateurs et par les personnes.

Normaliser le format des données

Les données doivent être formalisées sous un format standard.

Pour cela on utilise un graphe appelé **RDF** («Resource Description Framework ») qui est un modèle de données composé du triplet suivant :

- Le **sujet** (l'élément à décrire)
- Le **prédicat** (une propriété de cet élément)

- L'**objet** (la valeur de cette propriété, qui peut être un autre élément)

On peut les voir comme une phrase reliant 2 éléments, comme par exemple "U2 a écrit l'album 'Songs of Innocence'".

U2 est le sujet, 'a écrit l'album' est le prédicat et 'Songs of Innocence' est l'objet. Cet objet peut être le sujet d'un nouveau triplet, par exemple la date d'écriture : « Songs of Innocence est sorti en 2014 ».

Le but du web sémantique étant de relier des données du monde entier, ces 3 éléments doivent être uniques. Un RDF est donc un triplet d'**URI** (Uniform Resource Identifier). Le but principal d'une URI est justement de fournir un nom universellement unique à une ressource afin qu'il soit possible de lier des données de différentes sources partout dans le monde. Cela permet également d'ajouter une sémantique à l'information, c'est-à-dire que l'information est décrite dans l'URI.

RDF est un langage qui permet de stocker de l'information sous une certaine forme. Il existe plusieurs formats afin de sauvegarder les triplets RDF en un flux d'octets (**sérialisation**), les trois principaux étant **RDF/XML** (.rdf), qui est la syntaxe originale mais plus verbeuse que les autres, **N3** (.n3) et **Turtle**(.ttl), ce dernier étant un standard W3C. La plupart des outils savent jongler entre ces trois formats, notamment en reconnaissant l'extension.

Interroger une base de données

Lorsqu'une base de données est créée, il faut aussi savoir l'interroger pour récupérer des informations. Pour cela il faut interroger le serveur à l'aide d'un langage de requête. Chaque requête permet de chercher des informations dans les ontologies. Le langage le plus connu et le plus utilisé est SPARQL.

SPARQL permet principalement de chercher de l'information dans des bases de données structurées, d'où l'intérêt de structurer l'information.

Dans le cas d'un corpus de texte non structuré, il est également possible d'extraire de l'information, en utilisant le « machine learning ». On appelle cela le « **text mining** » ou « **text analytics** ». La traduction française moins communément utilisée est l'extraction de connaissances ou fouille de textes. Le text mining permet de transformer un texte non structuré en texte structuré par l'analyse d'une collection de ressources écrites. Il permet d'identifier les faits, les relations entre ces faits ainsi que les assertions. L'immense avantage de ce concept par rapport à la recherche par mots clefs (keyword search) est de permettre de reconnaître des concepts similaires même quand ils sont exprimés différemment.

Ici, on ne s'intéressera pas aux données non structurées.

Vocabulaire standard

On a dit que pour construire une base de données, celle-ci doit respecter le schéma RDF recommandé par W3C afin de construire un Web normalisé, compréhensible de tous. De même, il est également nécessaire d'avoir un langage standard, construit sur le modèle de données RDF. Ce langage, appelé **Web Ontology Language (OWL)** permet de définir des ontologies web structurées.

Toujours dans la perspective d'une utilisation universelle des données, des ontologies web écrites avec le standard OWL ont été créées pour être utilisées par tous. Cela permet d'éviter à chacun de créer tout un ensemble d'ontologies lors de la création d'une base de données, mais également cela permet à tout le monde d'avoir les mêmes définitions de concepts. C'est un peu la grammaire du web sémantique.

Voici une liste non exhaustive des principales ontologies, accompagnées du préfixe communément utilisé. À noter qu'il est possible de choisir n'importe quel préfixe étant donné que c'est une abréviation mais qu'utiliser les préfixes standards permet une compréhension plus universelle de son code.

- Le **FOAF**, littéralement Friend Of A Friend, permet de décrire des entités. Elle fournit des informations diverses sur des personnes ou choses (quelles qu'elles soient) ainsi que leurs relations.
foaf: <http://xmlns.com/foaf/0.1/>
- Le **Dublin Core** permet de décrire des ressources numériques ou physiques (titre, créateur, éditeur, sujet, langue, description, format, date ..)
dc: <http://purl.org/dc/elements/1.1/>
dct : <http://purl.org/dc/terms/>
Par exemple <http://purl.org/dc/elements/1.1/title> est le titre d'un document, d'un livre.
- Le vCard de W3 concerne le monde des affaires.
v : <http://www.w3.org/2006/vcard/>
Par exemple <http://www.w3.org/2006/vcard/title> est l'intitulé du poste d'une personne.
- Les relations entre objets liés au schéma RDF ainsi que la syntaxe des éléments sont décrites par les ontologies suivantes :
 - rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
 - rdfs: <http://www.w3.org/2000/01/rdf-schema#>
Exemple rdfs :label pour le nom d'une URI quelle qu'elle soit.
 - Pour le format d'un element.
xsd: <http://www.w3.org/2001/XMLSchema#>
Par exemple "4"^^xsd:integer

Où la donnée est-elle stockée ?

En théorie, on pourrait accéder à toutes les données du Web avec le langage standard SPARQL. Cependant ce n'est pas le cas pour le moment puisque seule une partie des fichiers du Web est standardisée au format RDF.

Néanmoins certaines personnes ou institutions ont créé des fichiers ayant pour vocation d'être utilisés universellement. Il s'agit donc en quelques sortes de bases de données de ressources et de propriétés libres. Ces fichiers respectent le schéma RDF ainsi que les standards OWL.

Ces systèmes de bases de données (SGBD) relationnelles (RDBMS-Relational Database Management System- en anglais) spécifiques au format des triplets sont appelées Triplestores et sont optimisés pour le stockage et l'extraction de triplets.

Quelques exemples de triplestores, dans un ordre quelconque: OpenLink Virtuoso, AllegroGraph, Stardog, Neo4J, MarkLogic.

Ces triplestores sont stockées sur internet pour être accessibles, à l'aide de **SPARQL endpoint**, ou point de terminaison. Cela fonctionne comme pour les moteurs de recherche classiques : on a une URL racine (« base URL ») et on lui ajoute les paramètres de la requête.

Si elles ne sont pas mises à jour, les bases de données deviennent obsolètes. Or mettre à jour une base de données prend du temps et ne peut pas être fait toutes les minutes. Sur chaque point de terminaison est stockée une base de données à une version v , à un instant t . Il est donc important de choisir un point de terminaison récent lorsqu'on souhaite interroger une base de données.

DBpedia possède différents SPARQL endpoints dont l'un est mis à jour régulièrement. Il s'agit de l'URL <http://live.dbpedia.org/sparql>. DBpedia live est le seul point de terminaison qui mette à jour régulièrement la base en ne mettant à jour que les données modifiées.

The screenshot shows the Virtuoso SPARQL Query Editor. At the top, there are navigation icons (back, forward, search) and a link to the live DBpedia endpoint. The main area has a blue header bar labeled "Virtuoso SPARQL Query Editor". Below it, a "Default Data Set Name (Graph IRI)" field contains "http://dbpedia.org". The "Query Text" section contains the following SPARQL query:

```
SELECT distinct *
WHERE {
?x a dbo:Event .
?x rdfs:label ?name .
BIND(STR(?name) as ?Subject) .
OPTIONAL{?x georss:point ?coordinates} .
?x dbo:place ?z .
OPTIONAL{ ?z rdfs:label ?nameobjectURI } .
BIND (COALESCE(STR(?nameobjectURI),str(?z)) as ?Object) .
OPTIONAL{?z georss:point ?place} .

BIND (concat("http:// wikipedia.org/wiki/",replace(?name," ","_")) as ?wikilink) .
FILTER (!bound(?nameobjectURI) )
}

LIMIT 10000
```

(Security restrictions of this server do not allow you to retrieve remote RDF data, see [details](#).)

Results Format:

Execution timeout: milliseconds (values less than 1000 are ignored)

Options:

- Strict checking of void variables
- Strict checking of variable names used in multiple clauses but not logically connected to each other
- Log debug info at the end of output (has no effect on some queries and output formats)
- Generate SPARQL compilation report (instead of executing the query)

(The result can only be sent back to browser, not saved on the server, see [details](#))

Voici d'autres SPARQL endpoints mais dont la mise à jour est en général bien antérieure à celle de DBpedia Live :

- <http://dbpedia.org/snorql/>
- <http://demo.openlinksw.com/sparql>
- <http://librdf.org/query/>

Il est normalement impossible d'interroger une base de données d'un autre point de terminaison que celui que vous utilisez sauf si vous incluez des informations à ce propos par exemple à l'aide de « SPARQL federation ». Ce dernier permet d'interroger plusieurs points de terminaison SPARQL en même temps pour obtenir un résultat combiné.

La base de données DBpedia

Le but de DBpedia est d'avoir une version structurée des données disponibles dans Wikipédia. L'équipe a mis au point un algorithme permettant de parcourir les pages de Wikipédia pour retrouver les informations brutes et les enregistrer dans la base.

Elle a été créée dans le but d'être utilisé par tous librement et automatiquement, notamment par les applications Web.

C'est cette base de données que nous allons interroger avec l'interface, puisque c'est l'une des plus riches existantes.

Organisation de l'interface

Pour la réalisation de mon interface comme pour la rédaction de mon rapport, j'ai choisi de définir certains mots de vocabulaire. Ce choix est évidemment discutable, notamment car j'ai appris de nouvelles choses au fur et à mesure du projet, mais je m'y suis tenue pour la compréhensibilité de mon rendu.

- J'ai appelé **élément** toute entité qui a une URI et donc une unicité. Il aurait peut-être été plus clair de choisir le mot objet mais celui-ci est déjà réservé pour le triplet RDF.
- J'ai appelé **sujet**, le sujet du triplet, donc la première partie du triplet.
- J'ai appelé **prédictat** le verbe du triplet. Celui qui permet en général d'obtenir l'objet.
- J'ai appelé **objet** la dernière partie du triplet. Celui-ci peut-être avoir une URI, donc un élément, ou un texte brut comme cela sera expliqué par la suite.

Guide utilisateur

L'interface comprend une page d'accueil qui explique rapidement et simplement à l'utilisateur comment utiliser l'application et quelle est son utilité. Cette explication est accompagnée d'exemples pour aider l'utilisateur à transformer correctement sa question.

The screenshot shows a user guide for a query application. The top navigation bar includes 'Let's query Wikipedia !' and a menu icon. The left sidebar has links for 'User Guide', 'Query', 'Map', and 'Further Information'. The main content area starts with a green header 'Welcome to baby Deep Thought !' followed by 'What is it for ?'. It explains that most people search on Wikipedia, but for more specific information like 'all the athlete who were born in France since 1950', it gets complicated. It states that users will easily get the results they want through this app. Below this is a section titled 'What you need to know before starting' which details how queries are split into triples (Subject, Predicate, Object) and provides examples. It also advises being careful with accents and uppercase. A list of instructions follows: click the Go button, click on Map, or download results. The next section, 'How does it work?', discusses DBpedia and SPARQL. At the bottom, there is a 'Examples' section with a link to 'Example 1'.

Requête

Ensuite l'onglet **Query** permet d'interroger la base de données. Au départ l'interface est au format le plus simple pour aider à la compréhension. Ensuite des menus déroulants apparaissent lorsque la sélection s'y prête. Il est également possible de choisir d'ajouter un prédictat.

Le fonctionnement de cet onglet est détaillé dans la partie suivante.

The screenshot shows the 'Query' interface with the following settings:

- Subject:**
 - Type of the subject: Person
 - Precision about the type of the subject: All
 - Value of the subject: optional
 - Exact match: unchecked
- Predicates and objects:**
 - Predicate: no
 - Precision: All
 - Object: Value of the object: optional
 - Exact match: unchecked
- Results:**
 - Number of results: A slider set to 200, with a scale from 0 to 10,000.
- Preferences:**
 - Go!
 - Download

Carte

En cas de requête avec résultats spatialisés vous pouvez les consulter sur une carte. Lorsque les 2 objets présentent une spatialisation, par exemple dans le cas du lieu de naissance et du lieu de mort, les 2 informations apparaissent, le deuxième objet étant affiché par un marqueur orange. En survolant le marqueur, le nom du sujet apparaît. En cliquant son nom apparaît ainsi que le nom du lieu. A cela s'ajoute le lieu de l'objet 2 si celui-ci est spatialisé.

See your results on a map !

In case of spatialized query, that is if coordinates appear in your table, a map is displayed with all located results on it.

How does it work ?

One blue marker for one result, so one line of the table ! If you have selected 2 spatialized predicates (ex: birth and death place), the second one will be displayed with orange markers The name of the subject will be displayed by hovering the marker.

Lot of people have several information, so for one result you can have both the country and the city.
If you have several markers for one element (Person, Organisation ..), you should select City or Country in the options.

See your results on a map !

In case of spatialized query, that is if coordinates appear in your table, a map is displayed with all located results on it.

How does it work ?

One blue marker for one result, so one line of the table ! If you have selected 2 spatialized predicates (ex: birth and death place), the second one will be displayed with orange markers The name of the subject will be displayed by hovering the marker.

Lot of people have several information, so for one result you can have both the country and the city.
If you have several markers for one element (Person, Organisation ..), you should select City or Country in the options.

Informations supplémentaires

L'onglet « Further information » permet de répondre aux problèmes qui sont souvent rencontrés, qui sont principalement dû à la base de données DBpedia elle-même. Je me suis pour cela basée sur ma propre expérience de l'interface puisque j'ai fait des centaines de tests pendant le développement, mais également sur les retours des personnes à qui j'ai fait tester l'interface.

De plus, si des questions demeurent, une partie « Contact » est prévue à cet effet.

De même des explications plus poussées sur le web sémantique ainsi que le code de l'application sont disponibles sur Github, via un lien dans le même onglet.

Fonctionnement de l'interface

Après avoir bien compris les concepts du web sémantique ainsi que le potentiel de DBpedia, j'ai pu apprendre à en interroger la base de données. Pour cela, j'ai appris à utiliser le langage SPARQL.

Dans cette partie je vais vous introduire le langage SPARQL en vous faisant découvrir ce qui se cache derrière l'interface.

Une requête SPARQL comprend, dans l'ordre :

- Une déclaration des préfixes qui permettent d'abréger les URI qui sont utilisées pour la définition des ontologies
- Une clause de résultat (SELECT) qui identifie les informations qui doivent être renvoyées
- La requête elle-même (WHERE) qui spécifie les données exactes qu'il faut interroger. Cette requête est formée de plusieurs clauses, organisées selon le triplet *sujet prédicat objet*.
- Des modificateurs de requêtes qui permettent d'ordonner ou de ne prendre qu'une partie des résultats.

Les préfixes utilisés pour l'interface sont les suivants :

```
PREFIX db: <http://dbpedia.org/resource/>
PREFIX dbp: <http://dbpedia.org/property/>
PREFIX dbo: <http://dbpedia.org/ontology/>
PREFIX dc: <http://purl.org/dc/elements/1.1/>
PREFIX dct: <http://purl.org/dc/terms/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
```

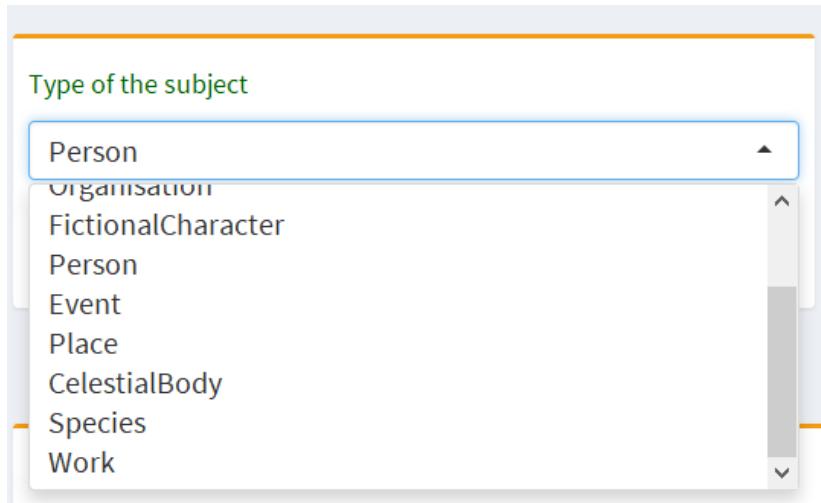
Par la suite les différentes possibilités de l'interface seront détaillées à l'aide d'exemples.

Certains des résultats des exemples présentés sont donnés en annexe.

Sujet

Type principal

On choisit le type d'ontologie qui nous intéresse.



Par exemple, on choisit de chercher des personnes, c'est la catégorie la plus fournie et la mieux remplie de DBpedia.

The screenshot shows the "Subject" section of the search interface. It contains two dropdown menus. The first dropdown is labeled "Type of the subject" and has "Person" selected. The second dropdown is labeled "Precision about the type of the subject" and has "All" selected. Both dropdowns have a downward-pointing arrow icon at the bottom right corner.

La requête correspondante est :

```
SELECT *
WHERE {
?x a dbo:Person .
?x rdfs:label ?y .
BIND (concat("http://wikipedia.org/wiki/",replace(?name," ","_")) as
?wikilink) .
}
LIMIT 100
```

La ligne correspondant particulièrement à ce qu'on est en train d'expliquer est en gras.

Remarques :

- ? permet de définir une variable.
- ?x a dbo:Person est équivalent à ?x rdf:type dbo :Person. Il s'agit d'un simple raccourci très pratique puisque ce prédicat est très souvent utilisé.
- ?x rdfs:label ?y . est la ligne permettant de créer un triplet même si on n'a que le sujet. En effet on part du principe qu'on a toujours besoin du nom de l'élément, donc le triplet est introduit automatiquement.
- BIND (concat("http://wikipedia.org/wiki/",replace(?name," ","_")) as ?wikilink) . permet l'ajout d'une colonne contenant le lien vers la page wikipédia de l'élément.

Type secondaire

On peut également vouloir préciser le type de sujet :

Voici les sous-catégories existantes :

Activity	Organisation	FictionalCharacter	Person	Event	Place	CelestialBody	Species	Work
All	All	All	All	All	All	All	All	All
Game	Company		Athlete		ArchitecturalStructure			Artwork
Sport	EducationalInstitution		Artist		NaturalPlace			Film
	Group		Politician		PopulatedPlace			MusicalWork
	SportsLeague		Writer					WrittenWork
	SportsTeam							

La requête correspondante est :

```

SELECT *
WHERE {
?x a dbo:Person .
?x a dbo:Athlete .
?x rdfs:label ?y .
BIND (concat("http://wikipedia.org/wiki/",replace(?name," ","_")) as ?wikilink) .
}
LIMIT 100
    
```

Dans le cas des lieux peuplés, c'est-à-dire les pays, régions, villes, villages, hameaux et autres, on peut ajouter une troisième précision : si on veut une ville ou un pays, les autres cas se trouvant dans « All ».

The screenshot shows a search interface with two main sections: 'Subject' and 'Object'. In the 'Subject' section, under 'Type of the subject', 'Place' is selected. Under 'Precision about the type of the subject', 'PopulatedPlace' is selected. Under 'Precision', 'City' is typed into a dropdown menu, with 'All', 'City', and 'Country' as other options. The 'Object' section is partially visible at the bottom right.

Précision du nom

On peut également vouloir préciser le nom de l'élément qui nous intéresse.

Par exemple, on veut l'athlète Teddy Riner :

```
SELECT *
WHERE {
?x a dbo:Person .
?x a dbo:Athlete .
?x rdfs:label "Teddy Riner"@en .
}
LIMIT 100
```

Autre exemple, toutes les villes contenant le mot Paris.

The screenshot shows a search interface with three main sections: 'Subject', 'Object', and 'Name of the subject'. In the 'Subject' section, 'Place' is selected as the type and 'PopulatedPlace' as the precision. In the 'Object' section, 'Paris' is typed into the 'Name of the subject' field. A checkbox for 'Exact match' is present but unchecked.

```
SELECT distinct *
WHERE {
?x a dbo:Place .
?x a dbo:PopulatedPlace .
?x rdfs:label ?name .
BIND(STR(?name) as ?Subject) .
FILTER(CONTAINS(?name,"Paris")) .
OPTIONAL{?x georss:point ?coordinates} .
BIND (concat("http://wikipedia.org/wiki/",replace(?name," ","_")) as
?wikilink) .
}
LIMIT 100
```

Remarque :

- *BIND(STR(?name) as ?Subject)* . permet de transformer l'URI en chaîne de caractères pour pouvoir l'étudier.

- Dans le cas de lieu, on ajoute les coordonnées GPS lorsqu'ils existent : *OPTIONAL{?x georss:point ?coordinates}*.

Pour Paris, comme pour d'autres éléments, il existe un grand nombre de résultats. Par exemple « Parish » signifie Paroisse en anglais, et en Estonie beaucoup de ville s'appelle « Paroisse de ... ». Il peut donc être intéressant de chercher le résultat exact dans ce cas là.

Subject		
Type of the subject	PopulatedPlace	Name of the subject
Place	Precision	Paris
	City	<input checked="" type="checkbox"/> Exact match

```

SELECT distinct *
WHERE {
?x a dbo:Place .
?x a dbo:PopulatedPlace .
?x rdfs:label ?name .
BIND(STR(?name) as ?Subject) .
FILTER(?Subject = STR("Paris"))
OPTIONAL{?x georss:point ?coordinates} .
BIND (concat("http://wikipedia.org/wiki/",replace(?name," ","_")) as
?wikilink)
}
LIMIT 100

```

De manière générale il vaut mieux éviter de cocher directement la case car il y a beaucoup d'écritures différentes qui peuvent être perdues. Par exemple « Paris, France » serait perdu. « America », en égalité stricte perdrait « United states of America », « American » ou toutes les variantes que les utilisateurs de Wikipédia ont choisis.

Le cas du sujet simple sert en général si on cherche une liste de personnes, ou une liste de lieux ... Dans la plupart des cas, on cherche des informations sur ces éléments, c'est l'intérêt des prédicats.

Prédicat et objet

Choix du prédicat

On choisit dans la liste qui nous est proposée. Cette liste varie en fonction du type de sujet qui a été sélectionné. Pour voir tous les prédicats par types de sujet, voir l'annexe « Prédicats ».

Subject	
Type of the subject	Precision about the type of the subject
Person	Artist
Predicates and objects	
Predicate	Object
Information you want about the subject	Name of the object
<input type="text" value="no"/>	<input type="text" value="optionnal"/>
no	<input type="checkbox"/> Exact match
birth name	
birth place	
birth date	
death place	
death date	
<input type="checkbox"/> nationality	
spouse	

Dans le cas de prédicats qui renvoient des objets spatialisés on peut, comme pour le sujet, préciser le type de lieu qui nous intéresse.

Predicate
Information you want about the subject
<input type="text" value="birth place"/>
Precision
All
All
City
Country

Voici un exemple avec un prédicat.

The screenshot shows a user interface for constructing a SPARQL query. It is divided into several sections:

- Subject** (top right):
 - Type of the subject: Person
 - Precision about the type of the subject: Writer
- Predicates and object** (center):
 - Predicate: Information you want about the subject (is author of)
 - Precision: All
- Object** (right side):
 - Name of the object: optional
 - Exact name: checked

```

SELECT distinct *
WHERE {
?x a dbo:Person .
?x a dbo:Writer .
?x rdfs:label ?name .
BIND(STR(?name) as ?Subject) .
?z dbo:author ?x .

OPTIONAL{ ?z rdfs:label ?nameobjectURI .}
BIND (COALESCE(STR(?nameobjectURI),str(?z)) as ?Object) .
BIND (concat("http://wikipedia.org/wiki/",replace(?name," ","_")) as
?wikilink) .
}
LIMIT 100
  
```

Deux remarques :

- Le prédicat donné est « is author of », le lien est donc indirect. Ici le sujet est le livre et l'auteur est l'objet. Dans un cas classique, le sujet de l'interface est sujet du triplet. Mais ceci se voit uniquement dans la requête et pas dans l'interface :
`?z dbo:author ?x .`
- Les objets renvoyés par DBpedia peuvent être des URI ou pas (les URI pouvant aller servir de sujet pour un triplet, alors qu'une chaîne de caractère par exemple ne le peut pas). Donc on transforme tout en chaîne de caractère pour un affichage uniforme :
`OPTIONAL{ ?z rdfs:label ?nameobjectURI .}
BIND (COALESCE(STR(?nameobjectURI),str(?z)) as ?Object) .`

Autre exemple, le lieu de naissance de Teddy Riner. Teddy Riner est le sujet, « est né à » est le prédicat et le lieu de naissance est l'objet.

```

SELECT *
WHERE {
?x a dbo:Person .
?x a dbo:Athlete .
?x rdfs:label 'Teddy Riner'@en .
?x dbo:birthPlace ?z .
OPTIONAL{ ?z rdfs:label ?nameobjectURI .}
BIND (COALESCE(STR(?nameobjectURI),str(?z)) as ?Object) .
BIND (concat("http:// wikipedia.org/wiki/",replace(?name," ","_")) as
?wikilink) .
}
LIMIT 100

```

Prédicat nommé- choix de l'objet

Dans les cas précédent, ce qui nous intéresse c'est de connaître l'objet. On connaît le sujet (Teddy Riner) et on voulait apprendre quelque chose sur lui à l'aide d'un prédicat (lieu de naissance).

Dans d'autres cas, on connaît l'objet et on cherche le sujet. Par exemple, on veut tous les athlètes nés en France.

```

SELECT distinct *
WHERE {
?x a dbo:Person .
?x a dbo:Athlete .
?x rdfs:label ?name .
BIND(STR(?name) as ?Subject) .
?x dbo:birthPlace ?z .
OPTIONAL{ ?z rdfs:label ?nameobjectURI .}
BIND (COALESCE(STR(?nameobjectURI),str(?z)) as ?Object) .
OPTIONAL{?z georss:point ?place} .
FILTER(CONTAINS(?Object,"France")) .
BIND (concat("http:// wikipedia.org/wiki/",replace(?name," ","_")) as
?wikilink) .
}
LIMIT 100

```

On peut le faire également avec des dates. Lorsqu'un prédicat permet d'obtenir des objets de type « date », une fenêtre apparaît pour choisir la période si l'utilisateur le souhaite. On peut entrer l'année minimale, maximale, les deux ou aucun. On peut également ne vouloir qu'une année, dans ce cas on entre l'année dans le champ « Name of the object » ou on l'écrit à la fois en minimum et maximum.

The screenshot shows the SPARQL query builder interface. In the 'Subject' section, the 'Type of the subject' is set to 'Event' and 'Precision about the type of the subject' is set to 'All'. In the 'Predicates and objects' section, the 'Predicate' dropdown is set to 'date' and 'Precision' is set to 'All'. The 'Object' section contains fields for 'Name of the object' (empty), 'minimum date (yyyy)' (1939), and 'maximum date (yyyy)' (1945). There is also an 'Exact match' checkbox for the name field.

```

SELECT distinct *
WHERE {
?x a dbo:Event .
?x rdfs:label ?name .
BIND(STR(?name) as ?Subject) .
OPTIONAL{?x georss:point ?coordinates} .
?x dbo:date|dbp:date ?z .
OPTIONAL{ ?z rdfs:label ?nameobjectURI .}
BIND (COALESCE(STR(?nameobjectURI),str(?z)) as ?Object) .
BIND(strdt(?Object,xsd:date) AS ?date).
FILTER(?date >= "1939-01-01"^^xsd:date).
FILTER(?date <= "1945-12-31"^^xsd:date).
BIND (concat("http://wikipedia.org/wiki/",replace(?name," ","_")) as
?wikilink) .
}
LIMIT 100

```

Deuxième prédicat

On peut parfois chercher à connaître plus qu'une information. C'est ce que permet la case suivante :

Add a new predicate

Lorsqu'elle est cochée, il est possible de choisir un deuxième prédicat et donc un deuxième objet.

The screenshot shows a user interface for constructing a SPARQL query. It has two main sections: 'Predicates and objects' and 'Object'.

Predicates and objects:

- Predicate:** Information you want about the subject. A dropdown menu shows 'no'. Below it, a 'Precision' dropdown shows 'All'.
- Add a new predicate:** A checked checkbox.
- Predicate 2:** Information you want about the subject. A dropdown menu shows 'no'. Below it, a 'Precision' dropdown shows 'All'.

Object:

- Name of the object:** An input field containing 'optionnal'.
- Exact match:** An unchecked checkbox.

Par exemple, on peut s'amuser un peu et chercher les dates de naissance et de mort de toutes les personnes s'appelant Smith.

```

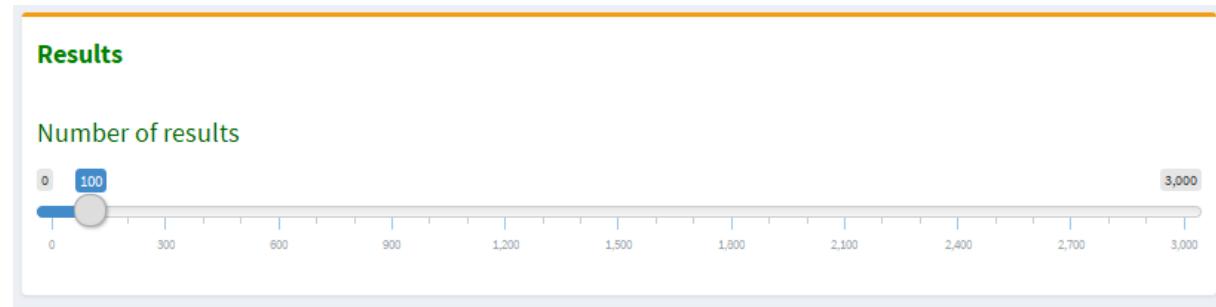
SELECT distinct *
WHERE {
?x a dbo:Person .
?x rdfs:label ?name .
BIND(STR(?name) as ?Subject) .
FILTER(CONTAINS(?name, "Smith")) .
?x dbo:birthPlace ?z .
OPTIONAL{ ?z rdfs:label ?nameobjectURI .}
BIND (COALESCE(STR(?nameobjectURI),str(?z)) as ?Object) .
OPTIONAL{?z georss:point ?place} .
?z a dbo:City .
?x dbo:deathPlace ?zbis .
OPTIONAL{ ?zbis rdfs:label ?nameobjectURIBIS .}
BIND (COALESCE(STR(?nameobjectURIBIS),str(?zbis)) as ?Object2) .
OPTIONAL{?zbis georss:point ?place2} .
?zbis a dbo:City .
BIND (concat("http://wikipedia.org/wiki/",replace(?name," ","_")) as
?wikilink) .
}
LIMIT 100

```

Résultats

Nombre de résultats

On peut bien entendu pour chaque requête, choisir le nombre de résultats souhaités. Cela correspond au « LIMIT n » que l'on trouve à la fin de chaque requête.



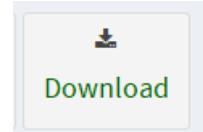
Par défaut, le nombre est de 100 car l'affichage est ainsi plus rapide et que 100 résultats suffisent dans les cas exploratoires.

Il est possible de choisir jusqu'à 3 000 résultats. Après avoir récupéré les résultats de Live DBpedia, ils sont traités sous R pour supprimer les doublons, souvent dû à de multiples coordonnées. Donc même si il est possible d'extraire jusque 10 000 résultats de la base de DBpedia, le nombre de résultats effectifs est inférieur, d'où la limitation à 3000 résultats (il y a en général 2 doublons par coordonnées).

Si l'on voulait les résultats suivant les 10 000 premiers résultats, un « OFFSET » dans la requête serait nécessaire. Je n'ai pas proposé cette fonctionnalité pour le moment car il est rare d'avoir un intérêt à avoir un si grand nombre de résultats.

Export

Il est possible d'exporter les résultats au format CSV pour pouvoir les réutiliser.



Spatialisation des résultats

Dans certains cas, les résultats peuvent être spatialisés. J'ai donc ajouté un item « Map » pour permettre de visualiser la localisation des résultats.

Il existe 3 cas :

- Le sujet est localisé. C'est le cas pour les lieux et les événements. Dans ce cas, même si l'objet est spatialisé on gardera uniquement les coordonnées du sujet car ce sont les plus précis.
- Un objet est localisé. C'est le cas lorsque le prédicat choisi est un lieu.
- Deux objets sont localisés. C'est le cas lorsque les deux prédicats choisis sont des lieux.

Visuellement, les deux premiers cas sont similaires puisqu'il y a une localisation par triplet.

Potentiels et limites

Problèmes liés à DBpedia

La base de données de DBpedia est remplie grâce à l'extraction automatique de données sur Wikipédia. C'est-à-dire que l'équipe en charge de son remplissage a mis au point un algorithme qui tente de récupérer la donnée brute dans le texte. Cela fonctionne généralement très bien mais il y a néanmoins de nombreux problèmes que l'interface ne peut pas gérer puisqu'ils viennent de la source.

Problème d'extraction de la nature

Par exemple voici les pages d'Usain Bolt sur Wikipédia et sur live.dbpedia.org, consultées le 5 février 2018.

https://en.wikipedia.org/wiki/Usain_Bolt

Usain Bolt

From Wikipedia, the free encyclopedia

"Usain" redirects here. For the organization with the acronym "USAIND", see United States Agricultural Information Network.

Usain St Leo Bolt OJ CD (/'ju:sɛn/) born 21 August 1986) is a retired Jamaican sprinter. He is the first person to hold both the 100 metres and 200 metres world records since fully automatic time became mandatory. He also holds the world record as a part of the 4 × 100 metres relay. He is the reigning Olympic champion in these three events. Due to his dominance and achievements in sprint competition, he is widely considered to be the greatest sprinter of all time.^{[9][10][11]}

An eight-time Olympic gold medalist, Bolt won the 100 m, 200 m and 4 × 100 m relay at three consecutive Olympic Games, although he subsequently forfeited one of the gold medals (as well as the world record set therein) nine years after the fact due to a teammate's disqualification for doping offences. He gained worldwide popularity for his double sprint victory at the 2008 Beijing Olympics in world record times. Bolt is the only sprinter to win Olympic 100 m and 200 m titles at three consecutive Olympics (2008, 2012 and 2016); this is a feat referred to as the "triple double" that will be very difficult for anyone to duplicate.

An eleven-time World Champion, he won consecutive World Championship 100 m, 200 m and 4 × 100 metres relay gold medals from 2009 to 2015, with the exception of a 100 m false start in 2011. He is the most successful athlete of the World Championships and was the first athlete to win three titles in both the 100 m and 200 m at the competition.

Bolt improved upon his second 100 m world record of 9.69 with 9.58 seconds in 2009 – the biggest improvement since the start of electronic timing. He has twice broken the 200 metres world record, setting 19.30 in 2008 and 19.19 in 2009. He has helped Jamaica to three 4 × 100 metres relay world records, with the current record being 36.84 seconds set in 2012. Bolt's most successful event is the 200 m, with three Olympic and four World titles. The 2008 Olympics was his international debut over 100 m; he had earlier won numerous 200 m medals (including 2007 World Championship silver) and holds the world under-20 and world under-18 records for the event.

His achievements as a sprinter have earned him the media nickname "Lightning Bolt", and his awards include the IAAF World Athlete of the Year, Track & Field Athlete of the Year, and Laureus World Sportsman of the Year (four times). Bolt retired after the 2017 World Championships, when he finished third in his last solo 100m race.^[12]

<p>Contents [hide]</p> <p>1 Early years</p> <ul style="list-style-type: none"> 1.1 Early competitions 1.2 Rise to prominence <p>2 Professional athletics career</p> <ul style="list-style-type: none"> 2.1 2004–2007 Early career 	 <p>The Honourable Usain Bolt OJ CD</p> <p>Bolt at the 2016 Summer Olympics</p> <p>Personal information</p> <p>Full name Usain St Leo Bolt Nickname(s) Lightning Bolt^[1] Nationality Jamaican Born 21 August 1986 (age 31) Sherwood Content, Jamaica Residence Kingston, Jamaica Height 1.95 m (6 ft 5 in)^[2] Weight 94 kg (207 lb)^[3]</p> <p>Sport</p> <p>Sport Track and field Event(s) Sprints Club Racers Track Club</p> <p>Achievements and titles</p> <p>Personal 100 m: 9.58 WR (Berlin 2009)^[4]</p>
---	---

On peut voir dès la première ligne la mention de « Jamaican sprinter », qui n'a pas été détectée dans Live DBpedia. Comme expliqué précédemment, on se base sur live DBpedia car il est mis à jour plus régulièrement. Le champs rdf:type est celui qu'on utilise pour savoir la nature d'un élément. On devrait y voir apparaître « dbo :Person » et « dbo :Athlete ».

① live.dbpedia.org/page/Usain_Bolt

The screenshot shows the DBpedia interface for the page of Usain Bolt. At the top, there is a navigation bar with the DBpedia logo, a 'Browse using' dropdown, a 'Formats' dropdown, and a link to 'dbc:IAAF_world_record_holders_(relay)'. Below this, the main content area has a heading 'rdf:type' followed by a list of categories under the owl:Thing namespace, such as yago:OlympicAthletesOfJamaica, yago:OlympicGoldMedalistsForJamaica, etc.

- owl:Thing
- yago:OlympicAthletesOfJamaica
- yago:OlympicGoldMedalistsForJamaica
- yago:OlympicMedalistsInAthletics(trackAndField)
- yago:PeopleFromTrelawnyParish
- yago:YagoLegalActor
- yago:JamaicanSprinters
- yago:LivingPeople
- yago:MaleSprinters
- yago:Runner110542888
- yago:PhysicalEntity100001930
- yago:Winner110782940
- yago:Athlete109820263
- yago:CausalAgent100007347
- yago:Contestant109613191
- yago:LivingThing100004258
- yago:Medalist110305062
- yago:Object100002684
- yago:Organism100004475
- yago:Person100007846
- yago:Sprinter110641413
- yago:Whole100003553
- yago:YagoLegalActorGeo

Mauvais classement de la donnée

De même, on peut avoir des éléments mal reconnus et qui n'apparaissent donc pas dans la bonne colonne de l'interface. Par exemple, la date de naissance d'Émile Rigaud apparaît dans la colonne normalement réservé au nom de naissance.

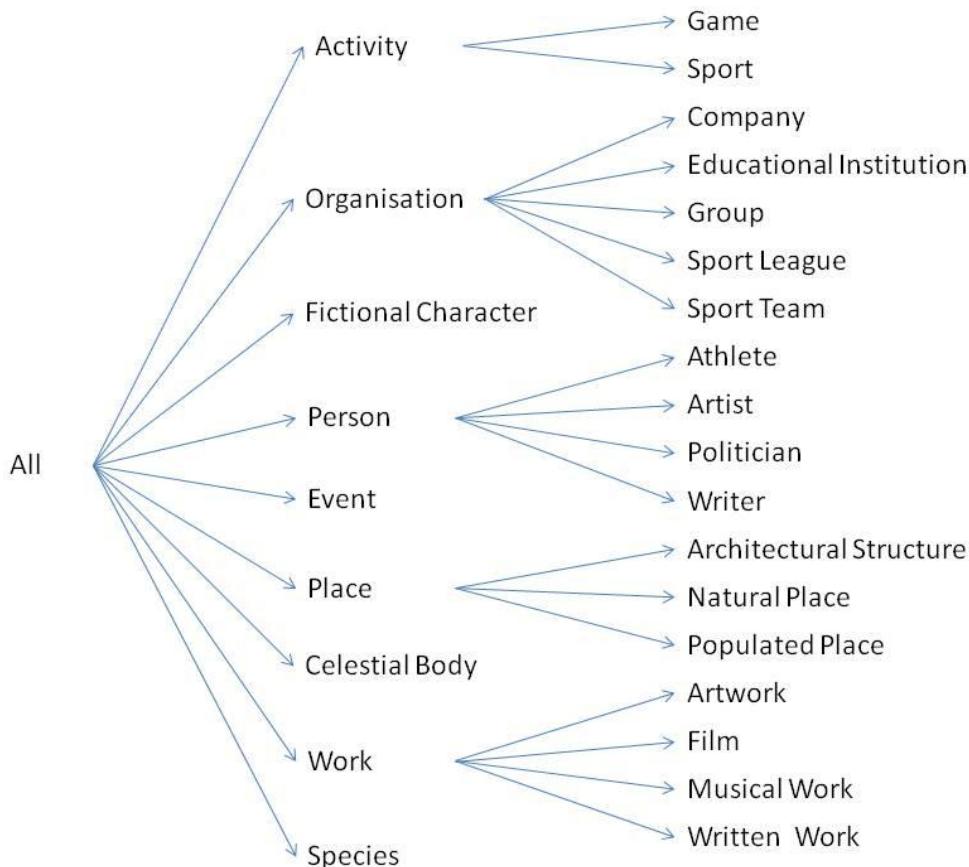
Pas de formatage de la donnée

Il n'y a pas de règle de formatage pour les éléments de DBpedia. Par conséquent on peut trouver des URI, tout comme des chaînes de caractères. On peut également trouver différentes écritures pour le même élément : Les États-Unis peuvent être appelé « United States of America », avec un « o » ou un « O », « America ». De même on pourrait trouver « Paris », « Paris,France » .. C'est pour cela que la case « Exact match » n'est pas cochée automatiquement dans l'interface. Elle permet juste d'aider l'utilisateur dans des cas où le nombre de résultats comprenant sa recherche est trop nombreux.

Choix d'extraction des ontologies

En raison des nombreux problèmes de « rdf:type », j'ai choisi de ne pas travailler avec l'ensemble des ontologies de DBpedia (listées ici : <http://mappings.dbpedia.org/server/ontology/classes/>).

Cela aurait pour majeur inconvénient de perdre l'utilisateur parmi toutes les natures que peuvent prendre un élément. J'ai donc choisi de choisir les natures suivantes :



Ce réseau est donc bien plus clair et ne prend pas en compte des natures en général très peu utilisées. Ces natures sont néanmoins accessibles en sélectionnant l'ensemble des types de sujets.

Cependant l'inconvénient de cette extraction est le statisme du schéma choisi. Si de nouvelles classes sont créées dans DBpedia (ce qui arrive rarement car nécessite un nouveau type de choses), elles ne seront pas prises en compte dans mon schéma. Cela nécessite donc un maintien à jour futur de ce document (format CSV).

De même j'ai choisi des prédictats pour chaque type d'élément, il n'est pas possible d'en récupérer d'autres. J'ai fait ce choix car il n'y a pas de liste exhaustive des prédictats possibles puisqu'on peut toujours en ajouter, cependant seuls quelques uns sont communs à la plupart des éléments d'une catégorie. Cela s'élève à 80 prédictats différents environ.

Malgré les ontologies de référence permettant de limiter le problème, il existe toujours plusieurs façons de décrire un élément. Par exemple, on trouve les ontologies dbp:nationality et dbo:citizenship pour décrire la nationalité d'une personne. Dans tous les cas similaires, c'est-à-dire ceux où il n'y a pas un prédicat majoritaire, j'ai choisi de mettre les deux, à l'aide de l'opérateur « OU » logique. Ainsi, le résultat sera rempli avec le prédicat non vide.

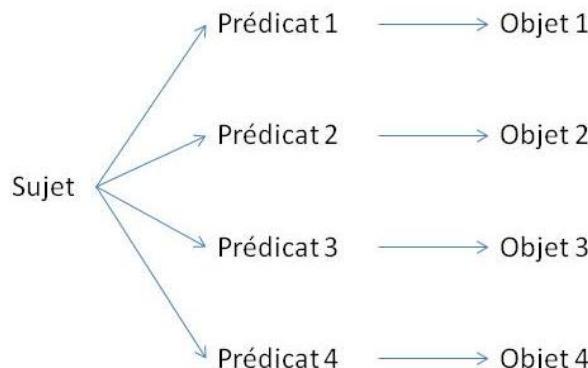
Tous les fichiers décrivant les ontologies et les prédicats choisis, ainsi que le dictionnaire créé pour une meilleure compréhension des prédicats dans l'interface sont disponibles en annexe au format CSV.

Choix des délimitations du projet

Les combinaisons possibles de triplets n'ayant de limite que l'imagination, il a été nécessaire de délimiter le projet.

Il existe principalement deux cas de figures d'enchaînement de triplets :

- Le premier consiste à construire les requêtes en arbre :



- Le second consiste à rendre l'objet sujet d'un nouveau triplet :



Ces deux cas de figure peuvent évidemment être tous les deux utilisés pour une même requête.

Etant donné que le format de la requête doit être automatisé, et que l'interface doit être organisée clairement, il n'est pas facile de laisser libre choix à l'utilisateur. De plus, entre autres problèmes techniques, un sujet nécessite d'être une URI alors qu'un objet ne l'est pas forcément. Par conséquent, le deuxième cas de figure peut se révéler difficile à traiter.

Enfin, nous avons estimé qu'avoir la possibilité d'utiliser deux prédicats est en général suffisante pour l'utilisateur et plus propice à donner des résultats (les prédicats ne sont pas toujours renseignés, voir « Potentiels et limites »). Néanmoins, il est possible de spécifier le nom de l'objet. Il s'agit donc d'un objet sujet d'un autre triplet, mais cela fonctionne uniquement pour ce prédicat.

Conclusion

En conclusion, le but final de ce projet de 5 semaines était de produire un outil permettant d'entrer une requête simple pour interroger DBpedia, à l'usage d'un utilisateur ne maîtrisant ni R ni SPARQL, et restituant le résultat de la requête sous forme tabulaire ou graphique.

Ce projet m'a permis d'apprendre beaucoup de choses puisque, à part le langage R appris cette année en cours, le reste était nouveau. J'ai découvert le web sémantique et le langage SPARQL , ce qui m'a beaucoup intéressée. J'ai également appris à créer une interface sous R avec la bibliothèque Shiny.

L'interface fournie permet cette interrogation de la base de données sans connaissances préalables. Elle pourrait par exemple être utilisée en complément de Wikipédia lors de recherches plus complexes. Le code est libre et disponible sur Github, tout comme l'application.

L'application est fonctionnelle et comprend déjà un certain nombre de fonctionnalités, il pourrait être intéressant d'aller au-delà des délimitations choisies pour le projet. On pourrait par exemple ajouter la possibilité de combiner les triplets, c'est-à-dire de faire de l'objet d'un triplet, le sujet d'un autre. On pourrait également le nombre de prédicats que l'on peut choisir. Il serait alors possible par exemple de chercher les personnes dont un parent est né aux Etats Unis avant 1950.

Si on veut utiliser cette application pour d'autres bases de données que celle de DBpedia, il faudrait changer le point de terminaison ainsi que les ontologies possibles. Pour le premier, c'est un simple changement d'URL, mais pour le second il faut créer les documents CSV délimitant les ontologies que l'on peut utiliser. Dans cette perspective on pourrait ajouter un module permettant de récupérer toutes les informations disponibles sur un élément. Dans ce cas seuls les sujets seront à définir, les prédicats nécessiteront un traitement après téléchargement.

Sitographie

Voici les URL des sites internet que j'ai utilisé, organisées par thèmes.

- Les bases du web sémantique :

<http://www.zeblogsante.com/web-3-0-definition/>

<http://infomesh.net/2001/swintro/>

<https://www.youtube.com/watch?v=r795n3AffgA&list=PL2ggTPNkhkw9sjvDDK4joBZpOc690DELI&index=2>

- Les langages de requêtes (SPARQL):

https://www.dropbox.com/s/zgvxvna132yhj1h/DuCharme_Learning%20SPARQL.pdf?dl=0

https://en.wikipedia.org/wiki/RDF_query_language

<https://www.youtube.com/watch?v=FvGndkpa4K0&index=3&list=PL2ggTPNkhkw9sjvDDK4joBZpOc690DELI>

- Le projet DBpedia:

<https://www.youtube.com/watch?v=BmHKb0kLGtA&index=4&list=PL2ggTPNkhkw9sjvDDK4joBZpOc690DELI>

- Les bases de données RDF :

<https://en.wikipedia.org/wiki/Triplestore>

- Ontologies et vocabulaire :

https://fr.wikipedia.org/wiki/Dublin_Core

https://fr.wikipedia.org/wiki/Web_Ontology_Language

- Le Text mining :

<https://archinfo41.hypotheses.org/571>

<https://www.youtube.com/watch?v=f7XN3RuDzmc&list=PL2ggTPNkhkw9sjvDDK4joBZpOc690DELI&index=1>

- Les endpoints :

<https://stackoverflow.com/questions/48229394/why-different-endpoints-do-not-query-same-datasets>

<https://bibliotheques.wordpress.com/2012/11/19/sparql-endpoint/>

<https://bibliotheques.wordpress.com/2012/06/14/sparql-premier-pas-installation-de-loutil-2/>

Annexes

Les annexes dans les pages qui suivent sont les suivantes :

Annexe 1 : Types des sujets

Type des sujets que l'on peut utiliser pour interroger DBpedia.

Annexe 2 : Liste des prédictats

Prédicats possibles selon le type du sujet.

Annexe 3 : Dictionnaire des prédictats

Pour chaque prédicat, l'ontologie correspondante ainsi que des informations sur son type (si le sens est direct ou non, si le prédicat donne des résultats spatialisés, si l'échelle est fixée)

Annexe 4 : Résultats de requête au format PDF

L'exemple utilisé est : « Les date et lieu de naissance des athlètes nommés Smith ».

Annexe 5 : GANTT

Détail de l'organisation des 5 semaines de projet

Types des sujets - Type des sujets que l'on peut utiliser pour interroger Dbpedia

Activity	Organisation	FictionalCharacter	Person	Event	Place	CelestialBody	Species	Work
All	All	All	All	All	All	All	All	All
Game	Company		Athlete		ArchitecturalStructure			Artwork
Sport	EducationalInstitution		Artist		NaturalPlace			Film
	Group		Politician		PopulatedPlace			MusicalWork
	SportsLeague		Writer					WrittenWork
	SportsTeam							

Liste des prédictats - Prédicats possibles selon le type du sujet

Person	Writer	Politician	Artist	Athlete	Place	PopulatedPlace	NaturalPlace
no	no	no	no	no	no	no	no
birth name	type of place	type of place	type of place				
birth place	place, localisation	place, localisation	place, localisation				
birth date	population	population	population				
death place	is birth Place of	is birth Place of	is birth Place of				
death date	is Part of	is Part of	is Part of				
nationality	nationality	nationality	nationality	nationality	is the city of	is the city of	total area (km ²)
spouse	spouse	spouse	spouse	spouse	total area (km ²)	total area (km ²)	
children	children	children	children	children			
parents	parents	parents	parents	parents			
type of job	title	term start	type of job	type of job			
team	greatest epoch	term end	title	team			
term start	is author of	party	greatest epoch	term start			
term end	style, genre	successor	starred in	term end			
party	has influenced	predecessor	is author of	title			
successor	was influenced by	title	style, genre	greatest epoch			
predecessor		greatest epoch	has influenced	was influenced by			
title		is author of	was influenced by				
greatest epoch		has influenced					
starred in		was influenced by					
is author of							
style, genre							
has influenced							
was influenced by							

Liste des prédictats - Prédicats possibles selon le type du sujet

ArchitecturalStructure	CelestialBody	Species	Work	WrittenWork	MusicalWork	Film	Artwork	Activity
no	no	no	no	no	no	no	no	no
type of place	gravity	family	author	author	author	author	author	necessary skills
place, localisation	period		release date	release date	release date	release date	release date	minimum number of players
is birth Place of	radius		owner	owner	country	country	owner	playing time for the game
is Part of	constellation		country	country	writer	writer		game minimum age
total area (km ²)	temperature		writer	writer	language	language		part of random chance
	absolute magnitude		language	language	director	director		game complexity
	apparent magnitude		director	style, genre	record label	starring		contact between players
			starring	publisher	album	producer		sport equipment needed
			producer			style, genre		playing region
			record label					type of sport
			album					category of sport
			style, genre					is the activity of
			publisher					

Liste des prédicts - Prédicats possibles selon le type du sujet

Game	Sport	Organisation	Company	EducationalInstitution	SportsTeam	SportsLeague
no necessary skills minimum number of players playing time for the game game minimum age part of random chance game complexity is the activity of	no contact between players sport equipment needed playing region type of sport category of sport is the activity of is the activity of	no country founding year chairman sport league is the team of manager city founded by type of organisation term start term end associated Band or Artist members of the band style, genre record label	no country founding year city founded by type of organisation	no country city founded by type of organisation is the team of	no country founding year chairman sport league manager city	no country founding year chairman sport league city

Liste des prédicats - Prédicats possibles selon le type du sujet

Group	FictionalCharacter	Event
no	no	no
term start	creator	place
term end	country	date
associated Band or Artist		
members of the band		
style, genre		
record label		

Dictionnaire des prédicts

subtitle	original	direct	unprecise_place	place
birth place	dbo:birthPlace	TRUE	TRUE	TRUE
starred in	dbo:starring	FALSE	FALSE	FALSE
is author of	dbo:author	FALSE	FALSE	FALSE
type of place	dbo:type dbp:status	TRUE	TRUE	TRUE
place, localisation	dbo:city dbo:location dbo:country dbp:country	TRUE	TRUE	TRUE
birth date	dbp:birthDate dbo:birthDate	TRUE	FALSE	FALSE
type of job	dct:description	TRUE	FALSE	FALSE
team	dbo:team	FALSE	FALSE	FALSE
term start	dbo:activeYearsStartDate dbp:termStart dbo:activeYearsStartYear	TRUE	FALSE	FALSE
term end	dbo:activeYearsEndDate dbo:activeYearsEndYear	TRUE	FALSE	FALSE
party	dbo:party	TRUE	FALSE	FALSE
successor	dbo:predecessor	FALSE	FALSE	FALSE
predecessor	dbo:predecessor	TRUE	FALSE	FALSE
title	dbp:title	TRUE	FALSE	FALSE
greatest epoch	dbp:years	TRUE	FALSE	FALSE
population	dbo:populationTotal	TRUE	FALSE	FALSE
is birth Place of	dbo:birthPlace	FALSE	FALSE	FALSE
is Part of	dbo:isPartOf	TRUE	FALSE	TRUE
is the city of	dbo:city	FALSE	FALSE	FALSE
total area (km ²)	dbo:areaTotal	TRUE	FALSE	FALSE
gravity	dbp:gravity	TRUE	FALSE	FALSE
period	dbp:period	TRUE	FALSE	FALSE
radius	dbp:radius	TRUE	FALSE	FALSE
constellation	dbp:constell	TRUE	FALSE	FALSE
temperature	dbp:temperature	TRUE	FALSE	FALSE
absolute magnitude	dbp:absmagV	TRUE	FALSE	FALSE
apparent magnitude	dbp:appmagV	TRUE	FALSE	FALSE
family	dbo:family	TRUE	FALSE	FALSE
author	dbo:author	TRUE	FALSE	FALSE
release date	dbp:date dbp:year dbo:releaseDate dbp:recorded	TRUE	FALSE	FALSE
owner	dbp:owner	TRUE	FALSE	FALSE

country	dbo:country dbp:country	TRUE	FALSE	TRUE
writer	dbo:writer dbo:artist	TRUE	FALSE	FALSE
language	dbp:language	TRUE	FALSE	FALSE
director	dbo:director	TRUE	FALSE	FALSE
starring	dbo:starring dbp:animator	TRUE	FALSE	FALSE
producer	dbo:producer	TRUE	FALSE	FALSE
record label	dbo:recordLabel	TRUE	FALSE	FALSE
album	dbo:album	TRUE	FALSE	FALSE
style, genre	dbo:genre	TRUE	FALSE	FALSE
publisher	dbo:publisher	TRUE	FALSE	FALSE
nationality	dbo:citizenship dbp:nationality	TRUE	FALSE	FALSE
spouse	dbo:spouse	TRUE	FALSE	FALSE
children	dbo:child	TRUE	FALSE	FALSE
parents	dbo:child	FALSE	FALSE	FALSE
death place	dbo:deathPlace dbp:deathPlace	TRUE	TRUE	TRUE
death date	dbp:deathDate	TRUE	FALSE	FALSE
birth name	dbp:birthName	TRUE	FALSE	FALSE
has influenced	dbo:influencedBy	TRUE	FALSE	FALSE
was influenced by	dbo:influencedBy	FALSE	FALSE	FALSE
necessary skills	dbp:skills	TRUE	FALSE	FALSE
minimum number of players	dbp:players	TRUE	FALSE	FALSE
playing time for the game	dbp:playingTime	TRUE	FALSE	FALSE
game minimum age	dbp:ages	TRUE	FALSE	FALSE
part of random chance	dbp:randomChance	TRUE	FALSE	FALSE
game complexity	dbp:complexity	TRUE	FALSE	FALSE
contact between players	dbp:contact	TRUE	FALSE	FALSE
sport equipment needed	dbp:equipment	TRUE	FALSE	FALSE
playing region	dbp:region	TRUE	FALSE	TRUE
type of sport	dbp:type	TRUE	FALSE	FALSE
category of sport	dbo:category dbp:category	TRUE	FALSE	FALSE
is the activity of	dbo:sport	FALSE	FALSE	FALSE
founding year	dbo:foundingYear	TRUE	FALSE	FALSE
chairman	dbo:chairman	TRUE	FALSE	FALSE

sport league	dbo:league	TRUE	FALSE	FALSE
is the team of	dbo:team	FALSE	FALSE	FALSE
manager	dbo:manager	TRUE	FALSE	FALSE
creator	dbo:creator	TRUE	FALSE	FALSE
place	dbo:place	TRUE	TRUE	TRUE
date	dbo:date dbp:date	TRUE	FALSE	FALSE
city	dbo:city dbo:hometown	TRUE	FALSE	TRUE
founded by	dbo:foundedBy	TRUE	FALSE	FALSE
type of organisation	dbo:type	TRUE	FALSE	FALSE
associated Band or Artist	dbo:associatedBand dbo:associatedMusicalArtist	TRUE	FALSE	FALSE
members of the band	dbp:bandMembers dbp:currentMembers	TRUE	FALSE	FALSE

Résultats de requête au format PDF

Person	birth place	birth date
Putter Smith	Bell, California	19/01/1941
Maggie Smith	Essex	28/12/1934
Robert Smith (musician)	Lancashire	21/04/1959
Amery Smith	Los Angeles	03/08/1964
Richard Smith (English guitarist)	Beckenham	12/12/1971
Robert Smith (musician)	Blackpool	21/04/1959
Maggie Smith	Ilford	28/12/1934
Jabbo Smith	Pembroke, Georgia	24/12/1908
Broderick Smith	Hertfordshire, England	17/02/1948
Frederic Marlett Bell-Smith	England	1846-09-26
Frederic Marlett Bell-Smith	London	1846-09-26
Major Bill Smith	Checotah, Oklahoma	21/01/1922
Nick Smith (milliner)	Liverpool	09/11/1979
Spencer Smith (musician)	Denver	02/09/1987
Jason Barry-Smith	Queensland	12/12/1969
O. C. Smith	Mansfield, Louisiana	21/06/1932
George Logie-Smith	Australia	02/12/1914
Johnny Smith	Birmingham, Alabama	25/06/1922
Jason Barry-Smith	Brisbane	12/12/1969
Dallas Smith (singer)	Canada	04/12/1977
Tommy Smith (saxophonist)	Edinburgh	27/04/1967
Nick Glennie-Smith	England	03/10/1951
Nick Smith (milliner)	England	09/11/1979
Paul Smith (fashion designer)	England	05/07/1946
T. V. Smith	England	05/04/1956
Michael Smith (performance artist)	Illinois	08/03/1951
Kyla-Rose Smith	Johannesburg	10/09/1982
Nick Glennie-Smith	London	03/10/1951
George Logie-Smith	Melbourne	02/12/1914
Kiki Smith	Nuremberg	18/01/1954
Sandy Smith	Scotland	02/08/1983
Jack Smith (artist)	United Kingdom	18/06/1928
Lonnie Liston Smith	United States	28/12/1940
O. C. Smith	United States	21/06/1932
Travis Smith (musician)	United States	29/04/1982
Rick Smith (Underworld)	Wales	25/05/1959
Michael Smith (performance artist)	Chicago	08/03/1951
O. C. Smith	Louisiana	21/06/1932
Johnny "Hammond" Smith	Louisville, Kentucky	16/12/1933
Kiki Smith	West Germany	18/01/1954
Chris Smither	Miami	11/11/1944
Chad Smith	Saint Paul, Minnesota	25/10/1961
Zak Smith	Syracuse, New York	16/07/1976
Paul Smith (rock vocalist)	County Durham	13/03/1979
Sandy Smith	Dunbar	02/08/1983
Adrian Smith	London Borough of Hackney	27/02/1957
Dallas Smith (singer)	British Columbia	04/12/1977
Lonnie Smith (jazz musician)	Lackawanna, New York	03/07/1942

Willi Smith	Philadelphia	29/02/1948
Robert Smithson	Passaic, New Jersey	02/01/1938
Tyler "Telle" Smith	Dayton, Ohio	09/08/1986
Todd Smith (musician)	Rockville, Maryland	03/12/1975
Lonnie Liston Smith	Richmond, Virginia	28/12/1940
Michael Peter Smith	South Orange, New Jersey	07/09/1941
Tony Smith (sculptor)	South Orange, New Jersey	23/09/1912
Jack Smith (artist)	Sheffield	18/06/1928
Wadada Leo Smith	Leland, Mississippi	18/12/1941
Michael W. Smith	Kenova, West Virginia	07/10/1957
Rick Smith (Underworld)	Carmarthenshire	25/05/1959
Elliott Smith	Omaha, Nebraska	06/08/1969
Cal Smith	Gans, Oklahoma	07/04/1932
Robert W. Smith (musician)	Daleville, Alabama	24/10/1958
Paul Smith (rock vocalist)	Billingham	13/03/1979
Jimmy Smith (musician)	Norristown, Pennsylvania	08/12/1925
T. V. Smith	Romford	05/04/1956
Paul Smith (fashion designer)	Beeston, Nottinghamshire	05/07/1946
Rick Smith (Underworld)	Ammanford	25/05/1959
Mike Smith (guitarist)	Middle River, Maryland	11/10/1973
Travis Smith (musician)	Bainbridge, Georgia	29/04/1982
Tab Smith	Kinston, North Carolina	11/01/1909
Dallas Smith (singer)	Langley, British Columbia (city)	04/12/1977
Fred Sledge Smith	Los Angeles, California	18/05/1933
Willi Smith	Pennsylvania	29/02/1948
Paul Smith (pianist)	San Diego, California	17/04/1922
Corey Smith (artist)	San Francisco, California	03/10/1977
Xanthus Russell Smith	Philadelphia, Pennsylvania	1839-02-26
Fred "Sonic" Smith	West Virginia	14/09/1948
Travis Smith (musician)	Georgia (U.S. state)	29/04/1982
Todd Smith (musician)	Maryland	03/12/1975

Résultats de requête au format PDF

wikilink	latitude	longitude
http://wikipedia.org/wiki/Putter_Smith	33.98	-118.18
http://wikipedia.org/wiki/Maggie_Smith	51.75	0.58
http://wikipedia.org/wiki/Robert_Smith_(musician)	53.80	-2.60
http://wikipedia.org/wiki/Amery_Smith	34.05	-118.25
http://wikipedia.org/wiki/Richard_Smith_(English_guitarist)	51.41	-0.02
http://wikipedia.org/wiki/Robert_Smith_(musician)	53.81	-3.05
http://wikipedia.org/wiki/Maggie_Smith	51.56	0.09
http://wikipedia.org/wiki/Jabbo_Smith	32.14	-81.62
http://wikipedia.org/wiki/Broderick_Smith	NA	NA
http://wikipedia.org/wiki/Frederic_Marlett_Bell-Smith	51.50	-0.12
http://wikipedia.org/wiki/Frederic_Marlett_Bell-Smith	51.51	-0.13
http://wikipedia.org/wiki/Major_Bill_Smith	35.47	-95.52
http://wikipedia.org/wiki/Nick_Smith_(milliner)	53.40	-2.98
http://wikipedia.org/wiki/Spencer_Smith_(musician)	39.76	-104.88
http://wikipedia.org/wiki/Jason_Barry-Smith	23.00	143.00
http://wikipedia.org/wiki/O._C._Smith	32.03	-93.70
http://wikipedia.org/wiki/George_Logie-Smith	-35.30	149.13
http://wikipedia.org/wiki/Johnny_Smith	33.65	-86.81
http://wikipedia.org/wiki/Jason_Barry-Smith	27.47	153.03
http://wikipedia.org/wiki/Dallas_Smith_(singer)	60.00	-95.00
http://wikipedia.org/wiki/Tommy_Smith_(saxophonist)	55.95	-3.19
http://wikipedia.org/wiki/Nick_Glennie-Smith	51.50	-0.12
http://wikipedia.org/wiki/Nick_Smith_(milliner)	51.50	-0.12
http://wikipedia.org/wiki/Paul_Smith_(fashion_designer)	51.50	-0.12
http://wikipedia.org/wiki/T._V._Smith	51.50	-0.12
http://wikipedia.org/wiki/Michael_Smith_(performance_artist)	41.28	-88.38
http://wikipedia.org/wiki/Kyla-Rose_Smith	26.20	28.05
http://wikipedia.org/wiki/Nick_Glennie-Smith	51.51	-0.13
http://wikipedia.org/wiki/George_Logie-Smith	-37.81	144.96
http://wikipedia.org/wiki/Kiki_Smith	49.45	11.08
http://wikipedia.org/wiki/Sandy_Smith	55.85	-4.27
http://wikipedia.org/wiki/Jack_Smith_(artist)	51.50	-0.12
http://wikipedia.org/wiki/Lonnie_Liston_Smith	40.72	-74.00
http://wikipedia.org/wiki/O._C._Smith	40.72	-74.00
http://wikipedia.org/wiki/Travis_Smith_(musician)	40.72	-74.00
http://wikipedia.org/wiki/Rick_Smith_(Underworld)	52.30	-3.60
http://wikipedia.org/wiki/Michael_Smith_(performance_artist)	41.84	-87.68
http://wikipedia.org/wiki/O._C._Smith	31.00	-92.00
http://wikipedia.org/wiki/Johnny_"Hammond"_Smith	38.25	-85.77
http://wikipedia.org/wiki/Kiki_Smith	50.73	7.10
http://wikipedia.org/wiki/Chris_Smither	25.78	-80.21
http://wikipedia.org/wiki/Chad_Smith	44.94	-93.09
http://wikipedia.org/wiki/Zak_Smith	43.05	-76.14
http://wikipedia.org/wiki/Paul_Smith_(rock_vocalist)	54.67	-1.83
http://wikipedia.org/wiki/Sandy_Smith	56.00	-2.52
http://wikipedia.org/wiki/Adrian_Smith	51.55	-0.06
http://wikipedia.org/wiki/Dallas_Smith_(singer)	54.00	-125.00
http://wikipedia.org/wiki/Lonnie_Smith_(jazz_musician)	42.82	-78.83

http://wikipedia.org/wiki/Willi_Smith	39.95	-75.17
http://wikipedia.org/wiki/Robert_Smithson	40.86	-74.13
http://wikipedia.org/wiki/Tyler_ "Telle" _Smith	39.76	-84.19
http://wikipedia.org/wiki/Todd_Smith_(musician)	39.08	-77.15
http://wikipedia.org/wiki/Lonnie_Liston_Smith	37.53	-77.47
http://wikipedia.org/wiki/Michael_Peter_Smith	40.75	-74.26
http://wikipedia.org/wiki/Tony_Smith_(sculptor)	40.75	-74.26
http://wikipedia.org/wiki/Jack_Smith_(artist)	53.38	-1.47
http://wikipedia.org/wiki/Wadada_Leo_Smith	33.41	-90.90
http://wikipedia.org/wiki/Michael_W._Smith	38.40	-82.58
http://wikipedia.org/wiki/Rick_Smith_(Underworld)	51.86	-4.31
http://wikipedia.org/wiki/Elliott_Smith	41.25	-96.00
http://wikipedia.org/wiki/Cal_Smith	35.39	-94.69
http://wikipedia.org/wiki/Robert_W._Smith_(musician)	31.30	-85.71
http://wikipedia.org/wiki/Paul_Smith_(rock_vocalist)	54.61	-1.27
http://wikipedia.org/wiki/Jimmy_Smith_(musician)	40.12	-75.34
http://wikipedia.org/wiki/T._V._Smith	51.58	0.18
http://wikipedia.org/wiki/Paul_Smith_(fashion_designer)	52.93	-1.22
http://wikipedia.org/wiki/Rick_Smith_(Underworld)	51.80	-3.99
http://wikipedia.org/wiki/Mike_Smith_(guitarist)	39.33	-76.44
http://wikipedia.org/wiki/Travis_Smith_(musician)	30.90	-84.57
http://wikipedia.org/wiki/Tab_Smith	35.27	-77.58
http://wikipedia.org/wiki/Dallas_Smith_(singer)	49.10	-122.66
http://wikipedia.org/wiki/Fred_Sledge_Smith	NA	NA
http://wikipedia.org/wiki/Willi_Smith	NA	NA
http://wikipedia.org/wiki/Paul_Smith_(pianist)	NA	NA
http://wikipedia.org/wiki/Corey_Smith_(artist)	NA	NA
http://wikipedia.org/wiki/Xanthus_Russell_Smith	NA	NA
http://wikipedia.org/wiki/Fred_ "Sonic" _Smith	NA	NA
http://wikipedia.org/wiki/Travis_Smith_(musician)	NA	NA
http://wikipedia.org/wiki/Todd_Smith_(musician)	NA	NA

