



Web Sémantique

Projet MovieRetriever

2 0 1 7

C O N T E N U

Ce rapport est une présentation de notre projet et la liste des spécifications pour celui-ci.

Par : Rémy AUDA et DRAVET Jean-Baptiste

Table des matières

Présentation projet	3
Description	3
Explication	3
Consignes	3
Ontologie	3
Les classes	3
Les propriétés d'objets	4
Le projet	4
Récupérateur de donnée	4
« Akinator »	4

Présentation projet

Le but de notre projet est de réaliser un logiciel qui permet de retrouver un film en répondant par oui ou par non à des questions qui seront posées.

Description

Le principe de notre logiciel est, via une interface graphique java, de poser des questions à l'utilisateur à propos d'un film afin de retrouver le titre du dit film. Pour cela, nous allons créer notre propre ontologie simplifiée.

Il s'agit d'un logiciel dans l'esprit de l'Akinator, à la différence qu'avec un Akinator, nous sommes en connaissance du film et ce serait à l'Akinator de le deviner alors qu'ici c'est l'inverse : la personne n'est pas en connaissance du film et notre logiciel essaye d'aider la personne à trouver le titre du film.

Explication

Nous avons décidé de ne pas récupérer une ontologie existante car elles nous ont semblé trop complexes avec beaucoup trop d'informations. En effet lorsque l'on recherche un film, on a souvent peu d'informations à son sujet. Poser des questions à propos de la musique ou des costumes nous a donc paru inutile. De plus, les grosses ontologies ont un système complexe d'ID (identifiant) qui lie les différentes classes entre elles, ce qui nous semble trop complexe à implémenter dans le temps imparti.

Consignes

Notre projet est lié aux consignes suivantes :

- RDF data
- RDFS / OWL / SKOS vocabularies, rules
- SPARQL queries (and show the inferences made, taking into account the semantics of the vocabulary)

Nous avons décidé de réaliser notre ontologie en OWL avec l'aide du logiciel Protege.

Ontologie

Voici une représentation de notre ontologie minimaliste. Cette ontologie est utilisée uniquement dans le but de notre logiciel. Elle se justifie par l'utilisation que nous en faisons.

Les classes

Voici la liste des différentes classes que nous avons créé :

- Actor : classe pour représenter les acteurs d'un film. Chaque acteur sera uniquement décrit par son nom, les films dans lesquels il a joué et le personnage qu'il a joué dans chaque film.
- Award : représente les récompenses pour les films.
- Characters : représente les personnages des films. Ils sont joués par des acteurs.
- Country : représente le pays d'origine d'un film.
- Movie : classe qui représente les films. En relation avec toutes les autres classes de l'ontologie.

- Producer : représente les producteurs pour chaque film. Ils font partie de la Team.
- Scriptwriter : représente les scénaristes pour chaque film. Ils font partie de la Team.
- Team : représente l'équipe humaine d'un film.
- Type : représente le type de film (le genre auquel il appartient).

Les propriétés d'objets

Voici la liste des différentes propriétés que nous avons créé :

- Characters : relation en un personnage et un film. Définit qu'un personnage appartient bien à un film.
- OriginalCountry : relation entre un film et son pays d'origine.
- hasActor : relation entre un film et un de ses acteurs. Inverse à la propriété hasPlayIn.
- hasPlayIn : relation entre un acteur et un des films dans lequel il a joué.
- hasScriptWriter : relation entre un film et son scénariste. Inverse de la propriété wasScriptWriterIn.
- wasScriptWriterIn : relation entre un scénariste et un de ses films.
- isTypeOf : relation entre un film et un de ses types (action, aventures, ...)
- wasPlayedBy : relation entre un personnage et l'acteur qui le joue.
- wasProducerIn : relation entre un producteur et un de ses films. Inverse de la propriété wasProductBy.
- wasProductBy : relation entre un film et son producteur.
- wasRewardWith : relation entre un film et une de ses récompenses.

Nous avons décidé d'ajouter un poids et une valeur booléenne pour chaque propriété et chaque valeur, ce poids et cette valeur seront expliqués dans la prochaine partie.

Le projet

Nous avons décidé de réaliser le projet en Java avec Corese pour la réalisation des requêtes SPARQL. Suite à la discussion avec notre intervenant, nous allons devoir diviser notre projet en deux projets bien distincts.

Récupérateur des données

Dans un premier temps, nous allons devoir créer un projet de récupération de données qui va, grâce au endpoint SPARQL, récupérer les données des bases de connaissances déjà existantes et remplir notre propre base de données. Pour se faire, il va falloir réaliser un mapping entre l'ontologie de la base de connaissance et la nôtre. Grâce au mapping et à des requêtes SPARQL, nous allons pouvoir remplir une base de données configurée pour notre propre ontologie.

« Akinator »

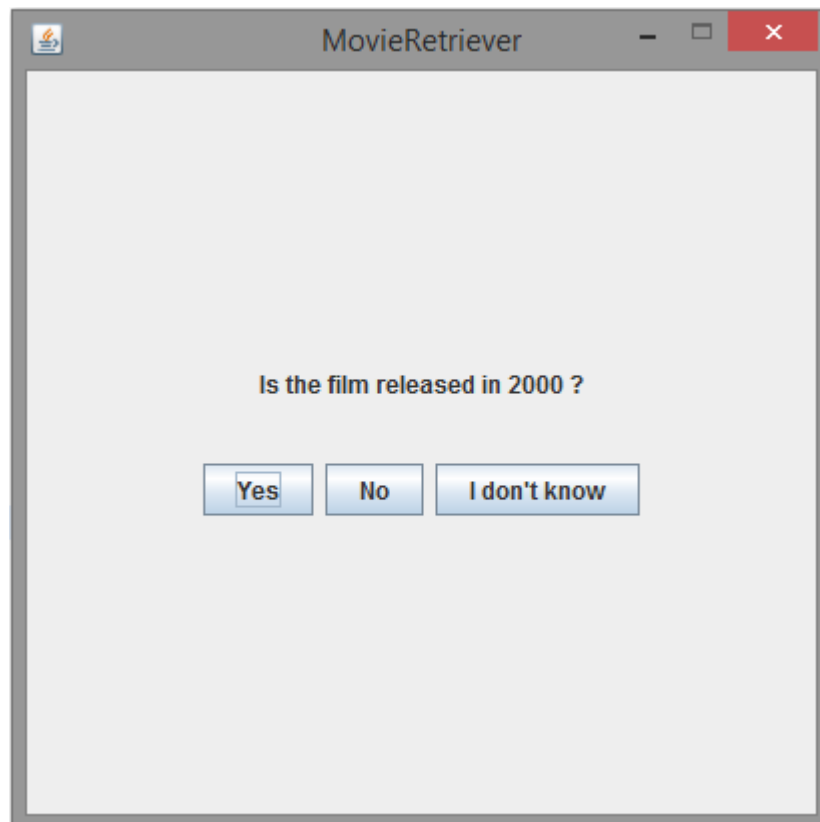
Une fois les données récupérées, nous allons lancer le second projet. Il s'agit d'un « Akinator » qui va poser des questions à l'utilisateur en utilisant notre ontologie et nos données. Pour cela, nous avons introduit un système de poids. Chaque valeur et chaque propriété ont un poids. Ce poids va nous permettre de définir l'ordre des questions et la sélection des valeurs. Les propriétés qui auront le plus gros poids seront les premières utilisées pour la réalisation des questions. Une fois utilisé, nous allons mettre à jour la base

de données en mettant une valeur à jour pour indiquer à notre algorithme que nous avons déjà utilisé cette propriété.

Les labels de chaque propriété et chaque valeur seront utilisés pour rédiger la question en langage naturel.

A chaque question, l'utilisateur va répondre par oui ou par non. Cette valeur nous permettra de créer au fur et à mesure la requête SPARQL jusqu'à ce que l'on n'obtienne qu'un film possible en réponse de cette requête. Tant que ce ne sera pas le cas, nous continuerons de poser des questions à l'utilisateur.

Une petite interface graphique (semblable à celle-ci-dessous) sera aussi développée pour éviter de devoir répondre dans la console.



Ainsi, au terme de ce projet, nous disposerons donc d'une application capable d'aider un utilisateur à retrouver un film dont il ne se souvient plus.

Auteurs : **AUDA Rémy** et **Jean-Baptiste DRAVET**