Titre : smart OER semantic scrapper

Le contexte est celui des ressources éducatives libres. En particulier celles organisées en France dans les grands catalogues des Universités Numériques. Le but est de pouvoir écrire, et surtout automatiser l'écriture) de *scrappers*, c'est à dire de logiciels qui vont pouvoir récupérer une ressource (un cours) avec les différents fichiers nécessaires (mais sans aller trop loin) de façon à pouvoir utiliser des outils d'analyse sémantique et comprendre ce qu'il y a dans le cours, comment il est organisé, etc.
Plus techniquement, il s'agit de préparer ces données afin de les transmettre à la plateforme X5-GON qui pourra alors les intégrer dans son catalogue.

Nombre d'étudiants : 2-3
Encadrants : Colin de la Higuera, Walid Ben Romdhane
Lieu : qui sait ?

Ci dessous, une description plus technique, en Anglais, du travail demandé.

Context:
In view of the current pandemic circumstances, the world has suddenly been pushed into an accelerated digital transition. This has led to many institutions/organizations to make earlier the needed migration of their systems.
In the education field as well, key actors like universities have been doing the same. Reinforced by the fact that Open Education is becoming a life philosophy in a very dynamic expansion, more institutions are sharing their OER (open educational resources) in well-structured and semi-structured repositories/catalogues through their OEC (open educational coursewares).

Description:
We want to automatically detect the semantic limits of an OER by using AI and machine learning approaches. Given a url or an OER we want to access that url and collect the content of that resource. On the road, the smart scrapper will find related materials (of several types) under "directly the same url" or a little bit further (level 1..n of the url hierarchy): the scrapper should be aware/intelligent of the limits of the scrapping operations.
In other words: While scrapping, define the frontiers of a given/searched OER

Guidelines:
Flexible architecture/algorithms to be able to accept "unsupervised" or even "semi-supervised" tasks by defining some configs/rules to find a quality OER.
It may be better to work in a group of 2/3 students to better dispatch tasks; here are some hints as indication:

- Prime Task: Design and set up the skeleton of the APP.
- First main task: Design and implement "system rules" (that can help the main ML algorithm to detect the correct contents)

- Second main task: the diving scrapping module (over the possible levels of the url hierarchy) in a clean manner(avoiding non-necessary content and dealing with the technical specifications meant for robots of the target website).
- Third main task: ML/semantic algorithms to decide if an OER should be added.
  ...

Keywords: OER, NLP, Machine learning, semantic web, ...

Technical stack:
Python, NLP & machine learning approaches.
Python/scrappy library
X5GON OER bank of resources

Skills:
python, data structure, efficient search algorithms, machine learning, NLP knowledge , ...
software design and programming best practices...

Use-case of the demo: the OER catalogue unit.eu

Support: first overview analytics done on the OERs found on "unit.eu"

Deliverable:
- A running soft with a read-me of how to install and run.
- Oriented service soft: could be an API or complete soft app.
- Flask API with X5gon specifs could be imposed : to be discussed later !!!!
- Demo with bench of examples.
- Evaluation demo/study/analytics on some reference examples (X5GON resources: html resources, or other).