TD - Information Extraction from Wikipedia pages

Download the .zip folder containing the lab exercise at http://moodle.i3s.unice.fr

Library for Python : NLTK

Write a program in the language of your choice. It is recommended to use a simple interpreted language such as Python.

1) Given as arguments:
- a set of Wikipedia articles, and
- a file with a list of entities and their URI, implement an algorithm that annotates the mentions of entities in the content of articles.

<entity name="http://en.wikipedia.org/wiki/Catherine_Deneuve">**Catherine Deneuve**</entity> (French: [katʁin dənœv]; born 22 October 1943) is a French actress who gained recognition for her portrayal of aloof, mysterious beauties for various directors, including Luis Buñuel and Roman Polanski. <entity name="http://en.wikipedia.org/wiki/Catherine_Deneuve">**She**</entity> is a twelve-time Cesar Award nominee.
<entity name="http://en.wikipedia.org/wiki/Catherine_Deneuve">**Deneuve**</entity> first came to prominence in Jacques Demy's 1964 film Les Parapluies de Cherbourg...

Your algorithm may fail in some cases, so verify if your entity recognition algorithm failed :
- By matching a subsequence of a word to one of the entities in the dictionary
- By recognizing a mention that refers to an entity not in the dictionary (ambiguous name)
- By not recognizing a mention that actually refers to one of the entities in the dictionary (a false negative).

2) Improve your algorithm, so that it uses a regular expression to find the birth date of the entity mentioned in the page.
Let it return a triple of the form <Entity, "hasDate", Date>. For example, it should return <"http://en.wikipedia.org/wiki/Catherine_Deneuve", "hasDate", "22 October 1943">
Try normalizing the dates you extract with your DateExtractor to the form YYYY-MM-DD
*Regular expressions in Python are implemented in the **re** module: https://docs.python.org/2/library/re.html*

3) Improve again your algorithm to extract the type of the article entity. For example, from a page starting with "**Catherine Deneuve** is a French actress", it should return the triple <"http://en.wikipedia.org/wiki/Catherine_Deneuve", "type", "French actress">. The subject of the triple is the uri of the detected entity, the predicate is always "type", and the object is what you extract.

4) Patterns extraction: Find patterns in Wikipedia, given seed pairs. For example,

* text: "Deneuve has been married to photographer David Bailey"
* pair: Deneuve, David Bailey
* pattern: "*x* is married to *y*"

Applying such pattern to all the Wikipedia pages of our corpus, return the triples:
<"http://en.wikipedia.org/wiki/Catherine_Deneuve", "marriedTo",
"http://en.wikipedia.org/wiki/David_Bailey">
<"http://en.wikipedia.org/wiki/Chiara_Mastroianni", "marriedTo", "
http://en.wikipedia.org/wiki/Benjamin_Biolay">
...

To be able to capture the language variability in expressing a certain relation, think about other patterns to express the same relation (e.g. X is the husband of Y for the relation “marriedTo”) and apply them to the corpus to retrieve additional triples.

RENDU TD: the algorithm, the obtained files, and a report (README file) where you explain how you have carried out the exercises and the results obtained.