M1 TAL - Data Science - UE 803

Exercise Sheet # 3

February 2, 2022

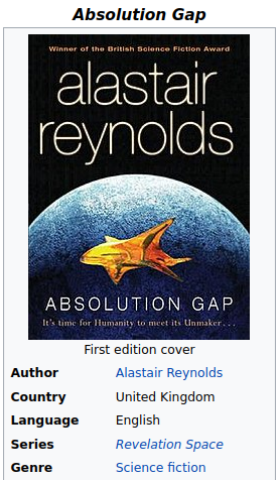
1 Introduction

In this practical session, you are asked to develop a python module which will collect data about books by scraping wikipedia pages.
You will need to :

1. find useful wikipedia articles (basically lists of books, such as `https://en.wikipedia.org/wiki/List_of_science_fiction_novels`, which will serve as starting points for your scraping)
2. download and parse these articles in order to extract the links to books they contain
3. for each such link, download the corresponding wikipedia page, and extract the following pieces of information and store them in a csv file (first manually inspect the HTML source code to locate these, and then use `BeautifulSoup` to extract them) :
   - title
   - author
   - genre
   - language
   - plot

Remarks

- The plot is usually contained in a dedicated section within the page (e.g. `<h2>Plot</h2>` or `<h2>Plot summary</h2>`) ;
- The other pieces of information are stored in so-called infoboxes (html tables) such as the following :




```
<tr>
<th scope="row" class="infobox-label">Author</th>
<td class="infobox-data">
<a href="/wiki/Alastair_Reynolds" title="Alastair Reynolds">
Alastair Reynolds</a></td>
</tr>
<tr>
<th scope="row" class="infobox-label">Country</th>
<td class="infobox-data">United Kingdom</td>
</tr>
```

2 Tips

Here are some information which may help you for each of the above steps.

2.1 Finding lists of books

To develop and test your module, you can pick among the lists contained in the following page :

`https://en.wikipedia.org/wiki/Lists_of_books`

Your code should at least support the following lists :

- `https://en.wikipedia.org/wiki/List_of_science_fiction_novels`
- `https://en.wikipedia.org/wiki/Great_Books_of_the_20th_Century`
- `https://en.wikipedia.org/wiki/List_of_films_considered_the_best`

2.2 Downloading web pages

Beware that some websites discourage (if not disallow programmatic http queries). In this context, it is good practice to define a user agent whenever using Python's `requests` library (`url` refers to a variable which should have been defined prior to invoking `requests`):

```
user_agent = {'User-agent': 'Mozilla/5.0'}
response   = requests.get(url, headers = user_agent)
```

2.3 Parsing web pages

To parse HTML pages with `BeautifulSoup`, recall there are two main search interfaces :

- via html elements (using e.g. `soup.find`)
- via css selectors (using e.g. `soup.select`)

Note that you can search by element and attribute-value pairs.
At some point, you may need to search for text in between to headers. To do so with `BeautifulSoup`, you may do as follows:

- Search for the beginning tag of the section your interested in ;
- Loop over all HTML tags until you fin the end tag :

```
for h in headers:
nextNode = h
    # Select the beginning tag
     while True:
            nextNode = nextNode.find_next_sibling()
            if nextNode is None: # no next node (end of the DOM)
                 break
            if isinstance(nextNode, NavigableString): #next node is a string
                 print (nextNode.strip())
            if isinstance(nextNode, Tag):  #next node is a tag
                 # check whether this is the end tag or not
```