ACTIVITAT 3

RESTREJADOR WEB AMB SCRAPY

24 D'OCTUBRE DE 2018

EDUARD BERENGUER VILELLA

ADRIÀ PEDRAZA SANTOS

1. EL RASTREJADOR

1.1 INTRODUCCIÓ

En aquesta pràctica el nostre objectiu ha estat el desenvolupament d'un rastrejador web, mitjançant la llibreria *scrapy* de *Python*, per tal d'extreure'n informació i indexar-la posteriorment amb l'*ElasticSearch*.

En el nostre cas, com que som uns apassionats del setè art, hem decidit indexar una col·lecció de pel·lícules juntament amb les seves característiques més rellevants. Per tal d'aconseguir-ho, el nostre *crawler* rastrejarà una pàgina web dedicada a la crítica cinematogràfica amb un catàleg molt ampli i que a part, ens permetrà *parsejar* tota la informació que necessitem.

1.2 RASTREIG I EXTRACCIÓ DE DADES

Iniciem el procés de rastreig per tal d'obtenir els termes que formaran part del nostre índex establint com a punt de partida la pàgina web següent: https://www.filmaffinity.com/en/topgen.php?genre=&fromyear=&toyear=&country=&nodoc&notvse

En aquesta pàgina, hi podem trobar un llistat de totes les pel·lícules que conté la web ordenades en funció de la seva puntuació. Cada element d'aquesta llista correspon a una pel·lícula diferent i conté la seva informació principal. D'aquesta informació, el rastrejador en parsejarà i n'extraurà el següent contingut: **títol**, **director**, **país de producció**, **puntuació** i **adreça URL** de la seva pàgina web individual en la qual hi podem trobar informació més detallada de la pel·lícula.

Un cop extreta la informació bàsica d'una pel·lícula, el rastrejador accedeix a la seva pàgina web individual utilitzant la URL que havíem obtingut abans i un cop allà, parsejem informació tal com: **la sinopsis**, **el gènere** al qual pertany, el nom dels **actors** que hi apareixen, **la duració** o l'**any** en que es va estrenar la pel·lícula.

Un cop extreta tota la informació de les pel·lícules de la llista indiquem al rastrejador una adreça en el qual tractar més elements, ja que en el nostre cas, la pàgina tant sols ens permetia visualitzar un llistat de 30 pel·lícules com a màxim. Finalment, aquest procés de rastreig i extracció es repeteix fins que ja no queden més elements que tractar.

1.3 PRINCIPALS MODIFICACIONS RESPECTE EL CODI D'EXEMPLE

Respecte el codi del rastrejador utilitzat a classe com a exemple, hem realitzat les modificacions següents:

- Hem afegit l'atribut **num_films** a la classe per tal de comptabilitzar el nombre de pel·lícules que han estat parsejades pel rastrejador. Aquest atribut junt amb la constant **MAX_FILMS** ens permeten controlar o definir el nombre de pel·lícules que parsejarà el rastrejador. Això ho hem fet així per tal d'agilitzar el procés d'extracció de dades ja que la col·lecció de pel·lícules de la pàgina es molt amplia i tampoc volíem que es demores excessivament les proves amb rastrejador. A part, aquest atribut també ens servirà posteriorment com a paràmetre per rastrejar més pel·lícules.

  Pe altra banda, com és obvi també hem modificat el valor dels atributs **start_url** i **allowed_domains** adaptant-los al nostre exemple. En el primer atribut, hem definit l'adreça en el qual el nostre rastrejador començarà el procés de rastreig i en l'últim, hem limitat el rastreig a adreces al domini *filmaffinity.com*.

- En la definició del mètode **Parse** de la nostra classe, hem utilitzat un bucle *for* al igual que en l'exemple per iterar sobre els segments de la pàgina o etiquetes HTML que obtenim del paràmetre *response* i que contenen la informació a extreure de cada pel·lícula. En el nostre cas però, aquesta informació quedava en dos etiquetes diferents i per tant, hem ajuntat els dos segments utilitzant la funció zip() on la variable *item* conté el segment amb les dades de la pel·lícula i la variable *mark*, el segment amb la puntuació.

  En relació al parseig i recuperació de les dades, el procediment ha estat similar al del exemple. S'ha parsejat la informació de la pàgina mitjançant el mètode *css()* i s'ha definit un diccionari amb la variable *doc*, on s'han anat guardant les parelles amb la informació extreta del parseig. La única diferència que podem trobar dins del codi del bucle, és la instrucció que incrementa el valor del atribut que comptabilitza el nombre de pel·lícules que s'han tractat. Un cop s'han realitzat totes les instruccions de parseig, es fa una crida a la funció **Parse_detail** en el que es recupera més informació i s'afegeix al diccionari.

  Fora del bucle *for* entrem al segment del codi on es comprova si s'han de tractar o no més elements. En cas afirmatiu, se li proporciona una nova adreça i s'executa de nou el mètode *parse* amb aquets nous *items*. A diferència de l'exemple fet a classe, la nostra pàgina no té cap element en el qual poder obtenir l'enllaç a una nova pàgina amb nous elements. Per aquest motiu, que hem explicat més detalladament en l'apartat de dificultats al implementar el rastrejador, en el nostre cas hem utilitzat el mètode *FormRequest* passant com a paràmetre el valor de l'atribut *num_films* per tal d'obtenir noves pel·lícules.

- En la definició del mètode **parse_detail**, no hi ha hagut masses diferències respecte a la implementació del rastrejador fet a classe, tret de la informació a recuperar en cada cas.

```
import scrapy

MAX_FILMS= 2000 # Constant that sets a limit of films to scrap

class TopfilmaffinitySpider(scrapy.Spider):
    name = 'TopFilmaffinity'
    allowed_domains = ['filmaffinity.com']
    web_url =
'https://www.filmaffinity.com/en/topgen.php?genre=&fromyear=&toyear=&country=&nodoc&notvse'
    start_urls = [web_url]
    num_films = 0

    def parse(self, response):

        for item, mark in zip(response.css('li.content'),
response.css('li.data')):

            doc = {}
            data = item.css('div.mc-info-container')
            doc['title'] = data.css('div.mc-title a::text').extract_first()
            doc['url'] = response.urljoin(data.css('div.mc-title
a::attr(href)').extract_first())
            doc['country'] = data.css('div.mc-title
img::attr(title)').extract_first()
            credits = data.css('div.mc-director')
            doc['director'] = credits.css('span.nb a::text').extract_first()
            doc['mark'] = mark.css('div.avg-rating::text').extract_first()
            self.num_films += 1

            yield scrapy.Request(doc['url'], callback=self.parse_detail, meta=doc)

        if self.num_films < MAX_FILMS:
            yield scrapy.FormRequest(url=self.web_url, formdata={'from':
str(self.num_films)}, callback=self.parse)


    def parse_detail(self, response):
        detail = response.meta
        data = response.css('dl.movie-info')
        detail['year'] =
data.css('dd[itemprop="datePublished"]::text').extract_first()
        detail['duration'] =
data.css('dd[itemprop="duration"]::text').extract_first()
        detail['cast'] = ' '.join(data.css('span.cast a span::text').extract())
        detail['genre'] = ' '.join(data.css('span[itemprop="genre"]
a::text').extract())
        detail['synopsis'] =
data.css('dd[itemprop="description"]::text').extract_first()

        yield detail
```

Implementació del rastrejador

1.4 RESULTATS OBTINGUTS PEL RASTREJADOR

En aquest apartat em utilitzat el rastrejador implementat anteriorment per tal d'obtenir uns resultats concrets, en el nostre cas les millors pel·lícules seleccionades per FilmAffinity.com . Sigui quin sigui la directiva indicada per comanda la informació que s'ens dóna de la pel·lícula resultant és la mateixa: títol, any d'estrena, url, director, país de producció, durada en minuts, la puntuació mitjana de la pel·lícula (donada pels usuaris de filmaffinity) i el gènere.

Totes aquestes **etiquetes** ens permeten diversificar molt el **tipus de pel·lícules** que es poden obtenir, per exemple, un usuari que li agradi el cinema clàssic usarà la directiva de l'any de producció més que un altre, un usuari que té especial interès en un director utilitzarà el nom d'aquest per trobar pel·lícules, en canvi ens podem trobar amb usuaris que solament vulguin obtenir les millors pel·lícules d'un gènere en concret.

A continuació em realitzat proves simulant els diferents tipus d'usuaris que poden usar aquest rastrejador:

Mencionar que per realitzar aquestes proves em de disposar de l'**ElasticSearch** executant-se en un terminal apart, així com ens em de situar en el directori del fitxer SearchIndex.py.

$ python3 SearchIndex.py --index film --query genre:drama
*Film és el nom del nostre índex
Aquesta consulta ens retorna totes les pel·lícules les quals siguin del gènere drama.

$ python3 SearchIndex.py --index film --query year:1994

Aquesta consulta ens mostra totes les pel·lícules que es van estrenar de cara al públic el 1994. És recomanable indicar més d'una any en qüestió quan l'any és massa antic degut a que no hi haurà pel·lícules avaluades d'aquell any.
La sortida d'aquesta comanda es pot veure en el fitxer **sortida.txt** .

$ python3 SearchIndex.py --index film --query director:Charles AND genre:Romance

Aquesta comanda ens retorna totes les pel·lícules del director "Charles" (Mencionar que em agafat aquest degut a que solament hi ha un director amb el nom de Charles, de cognom Chaplin) i que siguin del gènere romàntic.

$ python3 SearchIndex.py --index film --query year:2000 genre:Sci-Fi

Consulta que ens retorna tant les pel·lícules estrenades en l'any 2000 com del gènere de ciència ficció. El fet de no posar el "AND" en la consulta fa que el resultat obtingut sigui una concatenació dels dos resultats.

$ python3 SearchIndex.py --index film --query mark:8.5

Aquesta comanda ens retorna les pel·lícules que tenen com a puntuació donada pels usuaris de la pàgina web de 8.5 . En aquesta opció és recomanable indicar més d'una puntuació (hi ha puntuacions que no tenen resultats).

$ python3 SearchIndex.py --index film --query country:United AND genre:comedy

Aquesta comanda ens retorna com a resultat totes les pel·lícules que s'han rodat als Estats Units les quals siguin del gènere comèdia. En aquest cas no cal indicar-li el nom sencer del país de producció, ja que estan ben diferenciats.

Al realitzar qualsevol consulta també se'ns dóna informació del número de pel·lícules resultants obtingudes.

1.4.1 COMENTARIS

Comentar que ens hagués agradat que el rastrejador web fos capaç de poder indicar-li en una mateixa directiva més d'una paraula, per exemple:
Quan indicàvem el nom complet d'un director , per exemple director: Charles Chaplin, hagués estat molt bé que el resultat fos exactament totes les pel·lícules que tenen com a director Charles Chaplin, em pogut comprovar que el rastrejador solament agafa la primer paraula, en aquest cas Charles. (Em provat usant cometes '' i "" però no ha funcionat).

No ho considerem problemes com a tal, però podríem trobar molt pràctic el poder indicar-li al rastrejador d'una manera diferent com realitzar cerques específiques, per exemple, que usant l'operador '>' ens permetés trobar el llistat les pel·lícules les quals la seva duració és més gran que una certa quantitat de minuts o totes les pel·lícules que hi s'han estrenat a partir d'un any determinat.
D'aquesta manera donat un usuari exigent, que vol trobar pel·lícules amb unes característiques molt concretes (per exemple, que els anys de producció o les notes siguin molt variables, amb un director concret amb un gènere en concret) s'estalviaria fer consultes llargues amb molts "AND".

1.5 DIFICULTATS ALHORA D'IMPLEMENTAR EL RASTREJADOR

Les principals dificultats que ens hem trobat alhora d'implementar el nostre rastrejador han estat principalment aquelles derivades del parseig mitjançant el mètode d'extracció d'informació basat en CSS del *scrapy*. Amb aquest mètode es depèn completament de que la pàgina estigui ben estructurada i que les etiquetes HTML estiguin ben classificades. Concretament, ens hem trobat en alguna situació en el que volíem obtenir un segment de la pàgina i ens era impossible ja que l'etiqueta HTML en qüestió no tenia cap atribut *class*. En aquests casos, si tenien altres atributs les hem pogut filtrar cercant-les per la seva combinació atribut-valor. Pel que fa a les que no tenien cap mena d'atribut, ha estat impossible de filtrar-les.

Per altre banda, una de les coses que ens ha costat més solucionar ha estat trobar la forma de que el rastrejador realitzes la petició d'obtenir més pel·lícules. A diferència de l'exemple vist a classe, on podíem parsejar una etiqueta HTML que ens proporcionava una adreça a més TFG's, en el nostre cas, l'element de la pàgina que ens permetia fer això (un botó "load more results") no tenia cap adreça URL especifica. Tot i així, amb l'ajuda del inspector del Chrome a nivell de xarxa hem pogut observar com al clicar al botó "veure més resultats" es realitzava una petició a la mateixa URL de la pàgina amb el paràmetre "from:número" on el número indicava a partir de quin element s'havien de carregar els pròxims 30 elements de la llista de pel·lícules. Tal i com hem pogut observar anteriorment, això ho hem pogut solucionar en el nostre rastrejador utilitzant el mètode ***FormRequest***.

Finalment, una de les altres complicacions en la que ens hem trobat és que durant el procés de rastreig, arribats a cert punt, el servidor web al qual estem accedint per seguretat limita l'accés al nostre rastrejador degut al volum de peticions que fa i ens obliga a resoldre un *captcha* per tornar-hi accedir. Això provoca que no puguem rastrejar totes les pel·lícules de la col·lecció. Malgrat tot, com que podem extreure informació d'un volum de pel·lícules suficients per la realització d'aquesta pràctica, hem decidit obviar aquest problema. Tot i així, la llibreria *scrapy* proporciona eines per solventar-ho com la implementació de *delays* entre peticions, la rotació d'adreces IP que utilitza el rastrejador mitjançant *proxys* o la rotació del *user agent*.