ז׳אנה אילין

346186992

from selenium import webdriver

from selenium.webdriver.common.keys import Keys

from selenium.webdriver.support import expected\_conditions as EC

from selenium.webdriver.support.ui import WebDriverWait

from selenium.webdriver.common.by import By

from selenium.webdriver.common.action\_chains import ActionChains

from selenium.webdriver.support.ui import Select

import time

import os

from urllib.request import urlopen

import pandas as pd

from bs4 import BeautifulSoup

from urllib.request import urlopen

import requests

import json

def for\_gender(author):

for\_gen = author.split(' ')

name = for\_gen[0]

my\_url = 'https://api.genderize.io?name='+name

with requests.get(my\_url) as response:

response\_json = response.json()

return response\_json['gender']

def gender\_probability(author):

for\_gen = author.split(' ')

name = for\_gen[0]

my\_url = 'https://api.genderize.io?name='+name

with requests.get(my\_url) as response:

response\_json = response.json()

return response\_json['probability']

*# authorization on the site OpenAthens*

driver = webdriver.Firefox()

driver.maximize\_window()

driver.get('https://my.openathens.net/?passiveLogin=false')

button = driver.find\_element('xpath', '/html/body/app-root/ng-component/div/footer/div[1]/nav/ul[1]/li[1]/div/div/button[1]')

button.click()

time.sleep(4)

input\_nam = driver.find\_element('id', 'type-ahead')

input\_nam= input\_nam.send\_keys("Bar Ilan")

time.sleep(4)

button = driver.find\_element('class name', 'wayfinder-nav-link')

button.click()

time.sleep(4)

button = driver.find\_element('class name', 'ORRU02D-k-b')

button.click()

time.sleep(100). *#input login and password*

input\_search = driver.find\_element('id', 'search')

input\_search = input\_search.send\_keys('Karger')

time.sleep(3)

karger = driver.find\_element('xpath', '/html/body/app-root/ng-component/div/main/ng-component/app-page-body/div/app-research-view-desktop/div/div[1]/app-card-column/app-resources-card/lib-card/div/app-resources-view/div/ul/li/app-resource-content/a')

karger.click()

*#going to the Karger website and setting filters to search for research articles*

user\_agent = 'Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9.0.7) Gecko/2009021910 Firefox/3.0.7'

headers\_info={'User-Agent':user\_agent}

driver.get("https://karger-com.eu1.proxy.openathens.net/pps")

accept = driver.find\_element('xpath', '/html/body/div[1]/div[1]/div[2]/span[1]/a')

accept.click()

time.sleep(2)

content = driver.find\_element('xpath', '/html/body/section[1]/div[2]/div[2]/div/div[2]/nav/ul/li[1]/a')

content.click()

time.sleep(2)

articles = driver.find\_element('xpath', '/html/body/section[1]/div[2]/div[2]/div/div[2]/nav/ul/li[1]/ul/li[1]/a')

articles.click()

time.sleep(2)

checkbox = driver.find\_element('id', 'article-type--Research\_Articles-chk-0-0')

checkbox.click()

output = {'Paper title': [], 'paper DOI': [], 'Publication Date': [], 'Number of authors': [], 'Name of the first author': [], 'Name of the last author': [], 'Gender of the first author': [], 'Gender of the last author': [], 'First author gender probability': [], 'Last author gender probability': [], 'Affiliation of the first author': [], 'Affiliation of the last author': [], 'Number of Figures': [], 'Figure 1 caption': [], 'Figure 2 caption': [], 'Figure 3 caption': [], 'Figure 4 caption': [], 'Figure 5 caption': [], 'Figure 6 caption': [], 'Figure 7 caption': [], 'Figure 8 caption': [], 'Figure 9 caption': [], 'Figure 10 caption': [], 'Figure 1 Link': [], 'Figure 2 Link': [], 'Figure 3 Link': [], 'Figure 4 Link': [], 'Figure 5 Link': [], 'Figure 6 Link': [], 'Figure 7 Link': [], 'Figure 8 Link': [], 'Figure 9 Link': [], 'Figure 10 Link': [], 'Number of Tables': [], 'Table 1 caption': [], 'Table 2 caption': [], 'Table 3 caption': [], 'Table 4 caption': [], 'Table 5 caption': [], 'Table 6 caption': [], 'Table 7 caption': [], 'Table 8 caption': [], 'Table 9 caption': [], 'Table 10 caption': []}

n = 2 *#variable for Next button*

for page in range(104):

elements = driver.find\_elements('class name', 'sri-title')

for i in range(len(elements)): *# loop of articles*

counter\_start = 0

flag\_start = True

year\_limit = True

while flag\_start and counter\_start < 3:

try:

limit = driver.find\_elements('class name', 'al-citation-list')

if int(limit[i].text[22:26]) > 1997: *#check the year in the description of the article*

new\_elements = driver.find\_elements('class name', 'sri-title')

lnk = new\_elements[i].find\_elements('tag name', 'a')

lnk[0].click()

time.sleep(4)

flag\_start = False

else:

year\_limit = False

break

except:

driver.refresh()

flag\_start += 1

year\_limit = False

print('error in first try')

if year\_limit == False: *#if the year is less than 1998, go to the next article*

continue

for key in output: *#create dictionary key values ​​'not available'*

output[key] = output.get(key, []) + ['Not Available']

try:

time.sleep(4)

title = driver.find\_element('class name', 'wi-article-title')

output['Paper title'][-1] = title.text

DOI = driver.find\_element('class name', 'citation-doi')

output['paper DOI'][-1] = DOI.text

date =driver.find\_element('class name', 'article-date')

year =date.text[-4:]

output['Publication Date'][-1]= date.text

time.sleep(4)

try: *#**authors search*

name = driver.find\_element('class name', 'al-authors-list')

except:

try:

time.sleep(6)

name = driver.find\_element('class name', 'al-authors-list')

except:

name = False

if name:

all\_names = [i.strip(';') for i in name.text.split('\n') if i!=';']

num\_authors = len(all\_names)

first\_author = all\_names[0]

last\_author = all\_names[-1]

gen\_first = for\_gender(first\_author)

if not gen\_first:

prob\_first = 0.5

else:

prob\_first = gender\_probability(first\_author)

time.sleep(3)

gen\_last = for\_gender(last\_author)

if not gen\_last:

prob\_last = 0.5

else:

prob\_last = gender\_probability(last\_author)

time.sleep(3)

output['Number of authors'][-1] = num\_authors

output['Name of the first author'][-1] = first\_author

output['Name of the last author'][-1] = last\_author

output['Gender of the first author'][-1] = gen\_first

output['Gender of the last author'][-1] = gen\_last

output['First author gender probability'][-1] = prob\_first

output['Last author gender probability'][-1] = prob\_last

time.sleep(4) *# first affiliation*

nam\_aff = driver.find\_elements('class name', 'al-author-name')

first = nam\_aff[0]

but\_af = first.find\_element('class name', 'linked-name')

but\_af.click()

try:

time.sleep(4)

first\_affil = first.find\_element('class name', 'info-card-affilitation')

except:

time.sleep(5)

wait = WebDriverWait(first, 10)

first\_affil = wait.until(EC.presence\_of\_element\_located(('class name', 'info-card-affilitation')))

output['Affiliation of the first author'][-1] = first\_affil.text

but\_af.click() #

time.sleep(4) *# last affiliation*

aff = driver.find\_elements('class name', 'al-author-name')

last = aff[-1]

time.sleep(2)

but = last.find\_element('class name', 'linked-name')

but.click()

try:

time.sleep(5)

last\_affil = last.find\_element('class name', 'info-card-affilitation')

except:

try:

but\_af.click()

time.sleep(5)

but\_af.click()

time.sleep(5)

wait = WebDriverWait(last, 10)

last\_affil = wait.until(EC.presence\_of\_element\_located(('class name', 'info-card-affilitation')))

except:

print('\oops\n')

output['Affiliation of the last author'][-1] = last\_affil.text

time.sleep(5)

try*:* *#Figures and caption*

fig\_capt = driver.find\_elements('class name', 'fig-caption')

except:

time.sleep(5)

fig\_capt = driver.find\_elements('class name', 'fig-caption')

caption\_f = []

count\_f = 0

for l1 in fig\_capt:

if l1.text not in caption\_f and l1.text:

caption\_f.append(l1.text)

count\_f += 1

output['Number of Figures'][-1] = count\_f

for cap in range(len(caption\_f)):

output['Figure '+str(cap+1)+' caption'][-1] = caption\_f[cap]

time.sleep(4)

*#Save pictures*

fig\_l = driver.find\_elements('class name', 'download-slide')

link\_list =[]

counter = 0

for l2 in fig\_l:

ln = l2.get\_attribute('href')

time.sleep(5)

if ln not in link\_list and ln:

response = requests.get(ln, headers=headers\_info)

curr\_path = f"//Users//janny//Desktop//pictures//{year}"

curr\_path2 = f'//Users//janny//Desktop//pictures//{year}//{title.text}'

if not os.path.exists(curr\_path):

os.mkdir(f"//Users//janny//Desktop//pictures//{year}")

if not os.path.exists(curr\_path2):

os.mkdir(f'//Users//janny//Desktop//pictures//{year}//{title.text}')

with open(f"//Users//janny//Desktop//pictures//{year}//{title.text}//Fig{counter+1}.ppt", "wb") as f:

f.write(response.content)

link\_list.append(ln)

counter+=1

for link in range(len(link\_list)):

output['Figure '+str(link+1)+' Link'][-1] = curr\_path2

try: *#Tables and caption*

time.sleep(4)

tabl = driver.find\_elements('class name', 'table-wrap-title')

except:

time.sleep(6)

tabl = driver.find\_elements('class name', 'table-wrap-title')

count\_t = 0

cap\_tab = []

for l3 in tabl:

z = l3.find\_element('class name', 'caption')

cap\_tab.append(z.text)

count\_t+=1

output['Number of Tables'][-1] = count\_t

for num in range(len(cap\_tab)):

output['Table '+str(num+1)+' caption'][-1] = cap\_tab[num]

driver.back()

except:

driver.back()

print(title.text)

*#button “Next” for next page*

button = driver.find\_element('css selector', f'.pagination-bottom-outer-wrap > div:nth-child(1) > button:nth-child({n})')

link\_not\_work = True

times\_to\_try = 0

while link\_not\_work and times\_to\_try < 3:

try:

time.sleep(10)

button.click()

link\_not\_work = False

except:

print('error in third try')

driver.refresh()

times\_to\_try += 1

n=3 #*variable for Next button*. *From page 2 always n=3*

time.sleep(8)

driver.refresh()

df= pd.DataFrame.from\_dict(output)

df.to\_excel("/Users/janny/output.xlsx")