پروژه بازیابی اطلاعات

زهرا احمدی

سوال 1:

- در سوال 1 قسمت اول با استفاده از روش زیر crawler نوشتم:
- ابتدا HTMLSession , csv , nest_asyncio را IMPORT میکنیم.
- سپس در قسمت ارکایو سایت عصر ایران تاریخ را فیلتر کرده و از URL آن در searchURL استفاده میکنیم.
- سپس (news_nav.news_id_c. و div.header_pdate و h1.title a[title] و div.news_path a و div.subtitle) را از طریق کد :

پیدا میکنیم و سپس از طریق روش زیر آن را داخل یک فایل CSV میریزیم:

- writer = csv.writer(file)

```
from requests_html import HTMLSession
import csv
import nest_asyncio

nest_asyncio.apply()
baseURL = 'https://www.asriran.com'
searchURL = baseURL + '/fa/archive?service_id=-1&sec_id=-1&cat_id=-1&rpp=4000&from_date=1401/01/01&to_date=1401/01/28&p=1'
```

```
asession = HTMLSession()
r =  asession.get(searchURL)
resp=r.text
#print(resp)
```

Asyinc، text کل html است و با with open فقط اطلاعات هر خبرو میخونیم و با r=session.get... لینک مربوط به هر خبر و میگیرم و باز هم html است.

و بعد از html اون فیلدهایی که میخوایم و میکشیم بیرون و بعدش write میکنیم.

```
news = r.html.find('.vizhe_title')
# with open('dataset.csv', 'w') as f:
#     writer = csv.writer(f)
with open('dataset.csv', 'w', encoding='utf-8-sig', newline='') as file:
    for item in news:
        data=[]
        itemNews = HTMLSession()
        r = session.get(str(item.absolute_links)[2:][:-2])
        respNews = r.text
        strParagraph=''

        data.append(r.html.find('.news_nav.news_id_c')[0].text)
        data.append(r.html.find('div.header_pdate')[0].text)
        data.append(r.html.find('h1.title a[title]')[0].text)
        data.append(r.html.find('div.news_path a')[1].text)

        if len(r.html.find('div.subtitle')) != 0:
            data.append(r.html.find('div.subtitle')[0].text)
        else:
            data.append('')

        if len(r.html.find('div.body p')) != 0:
            data.append(r.html.find('div.body p')[0].text)
            for par in r.html.find('div.body p'):
                strParagraph += par.text
            data.append(strParagraph)
        else:
            data.append('')

        if len(r.html.find('a.link_en')) != 0:
            data.append(r.html.find('a.link_en')[0].text)
        else:
            data.append('')

        if len(r.html.find('divtags_title')) != 0:
            data.append(r.html.find('divtags_title'))
        else:
            data.append('')

        writer = csv.writer(file)
        writer.writerow(data)
```

پس از نوشتن کرولر و استخراج اطلاعات سوالات را پاسخ میدهیم

در سوال اول ابتدا Elastic Search را Loadمیکنیم. سپس ping میگیریم که مطئن بشیم متصل است.

```
from elasticsearch import Elasticsearch, helpers
import csv

es = Elasticsearch("http://localhost:9200")
print(es.ping())
```

```
True
```

در سوال 1 قسمت دوم (سوال 4) با استفاده از روش زیر dataرا در elasticsearch، index کردم و روی آن query زدم:

سپس ParsiAnalyzer-7.13.1 را هم نصب کرده و سپس از طریق کد زیر فایل csv ای که از مرحله قبل استخراج نموده ایم را index میکنیم:

برای قسمت index کردن فایل csv از کد زیر استفاده میشود:

```
# with open('NewData.csv',encoding="utf8") as f:
#     reader = csv.DictReader(f)
#     helpers.bulk(es, reader, index='news')

import pandas as pd
import json

df = pd.read_csv("NewData.csv",encoding="utf8")
json_str = df.to_json(orient='records')

json_records = json.loads(json_str)

es = Elasticsearch()
index_name = 'news'
doctype = 'census_record'
es.indices.delete(index=index_name, ignore=[400, 404])
es.indices.create(index=index_name, ignore=400)
action_list = []
for row in json_records:
    record ={
        '_op_type': 'index',
        '_index': index_name,
        '_type' : doctype,
        '_source': row
    }
    action_list.append(record)
helpers.bulk(es, action_list)
```

برای سوال 1 قسمت ب : سپس برای سوال اول که جایگاه کلمات در متن را بدهد به شرح زیر عمل کردم:

ابتدا از کاربر یک کلمه میگیریم و سپس با کد زیر کلمه را داخل متن خبر سرچ میکنیم:

```
word = input("Please Enter a word :")
res = es.search(index="news", body={"query": {"match_phrase": {"Body" : word}}})['hits']['hits']

for item in res:
    strWord = item['_source']['Body']
    print(strWord.find(word))
```

```
Please Enter a word :متخلفان
118
261
730
```

سپس برای سوال دوم که باید کد خبر های دارای تارخ 12-13 را بدهد :

با persian.convert... تمام فیلد های عددی را به انگلیسی تبدیل میکنیم تا بتوانیم روش سرچ بزنیم و بعد میایم روش find میزنیم.

```
import persian
def es_iterate_all_documents(es, index, pagesize=250, **kwargs):
    offset = 0
    while True:
        result = es.search(index=index, **kwargs, body={
            "size": pagesize,
            "from": offset
        })
        hits = result["hits"]["hits"]
        # Stop after no more docs
        if not hits:
            break
        # Yield each entry
        yield from (hit['_source'] for hit in hits)
        # Continue from there
        offset += pagesize

for entry in es_iterate_all_documents(es, 'news'):
    numPersian = persian.convert_fa_numbers(entry['Pdate'])
    if(numPersian.find('12') != -1):
        print(entry['News Code'],'-',entry['Pdate'])
    elif(numPersian.find('13') != -1):
        print(entry['News Code'],'-',entry['Pdate'])
```

که خروجی زیر را میدهد:

```
کد خبر ۸۳۵۶۶۱ - تاریخ انتشار : ۱۴۰۱-۰۱-۲۸ - ۱۳:۲۲
کد خبر ۸۳۵۶۳۱ - تاریخ انتشار : ۱۴۰۱-۰۱-۲۸ - ۱۲:۱۸
کد خبر ۸۳۵۶۳۲ - تاریخ انتشار : ۱۴۰۱-۰۱-۲۸ - ۱۲:۱۸
کد خبر ۸۳۵۵۲۳ - تاریخ انتشار : ۱۴۰۱-۰۱-۲۸ - ۱۲:۱۵
کد خبر ۸۳۵۵۵۹ - تاریخ انتشار : ۱۴۰۱-۰۱-۲۸ - ۱۳:۱۴
کد خبر ۸۳۵۵۵۱ - تاریخ انتشار : ۱۴۰۱-۰۱-۲۸ - ۱۳:۵۶
کد خبر ۸۳۵۵۲۸ - تاریخ انتشار : ۱۴۰۱-۰۱-۲۸ - ۱۳:۵۰
کد خبر ۸۳۵۵۱۹ - تاریخ انتشار : ۱۴۰۱-۰۱-۲۸ - ۱۳:۵۰
کد خبر ۸۳۵۵۴۹ - تاریخ انتشار : ۱۴۰۱-۰۱-۲۸ - ۱۳:۴۲
کد خبر ۸۳۵۵۴۶ - تاریخ انتشار : ۱۴۰۱-۰۱-۲۸ - ۱۳:۳۸
کد خبر ۸۳۵۵۴۸ - تاریخ انتشار : ۱۴۰۱-۰۱-۲۸ - ۱۳:۳۸
کد خبر ۸۳۵۵۳۷ - تاریخ انتشار : ۱۴۰۱-۰۱-۲۸ - ۱۳:۳۱
کد خبر ۸۳۵۵۲۱ - تاریخ انتشار : ۱۴۰۱-۰۱-۲۸ - ۱۳:۳۰
کد خبر ۸۳۵۵۴۷ - تاریخ انتشار : ۱۴۰۱-۰۱-۲۸ - ۱۳:۲۹
کد خبر ۸۳۵۵۴۵ - تاریخ انتشار : ۱۴۰۱-۰۱-۲۸ - ۱۳:۲۳
کد خبر ۸۳۵۵۴۴ - تاریخ انتشار : ۱۴۰۱-۰۱-۲۸ - ۱۳:۲۰
کد خبر ۸۳۵۵۱۴ - تاریخ انتشار : ۱۴۰۱-۰۱-۲۸ - ۱۳:۲۰
کد خبر ۸۳۵۵۴۳ - تاریخ انتشار : ۱۴۰۱-۰۱-۲۸ - ۱۳:۱۹
کد خبر ۸۳۵۵۴۰ - تاریخ انتشار : ۱۴۰۱-۰۱-۲۸ - ۱۳:۱۶
```

سوال2:

ابتدا elasticsearch را متصل میکنیم و ping میگیریم که مطمئن بشیم وصله یا نه.

سپس ادرس اصلی فایل ها را میدهیم و سپس با لیستشون و در میارم بعدش با اضافه کردن مسیر اصلی به اول هر فایل لیست فایل ها رو در میاریم.

```
from elasticsearch import Elasticsearch
import pandas as pd
import os
import re
import json


client = Elasticsearch(
    "http://localhost:9200"
)

print(client.ping())
```

سپس فایل های گرفته شده را index میکنیم

```
# Create index Question file
dir_path = r'C:\\App\\Data Files\\Question-Copy\\'

files_list=os.listdir(dir_path)
files_list=['C:\\App\\Data Files\\Question-Copy\\'+x for x in files_list]

data = pd.concat(map(pd.read_csv, files_list))
data = data.fillna("")
data
```

سپس خروجی به شکل زیر است:

	Id	PostTypeId	AcceptedAnswerId	ParentId	CreationDate	DeletionDate	Score	ViewCount	Body	OwnerUserId	...	LastEditDate	l
0	44899821	1	44900290.0		2017-07-04 07:39:23		-45	752	I am trying to fetch data from Android to C...	7800558.0	...	2017-07-04 09:58:45	
1	32004289	1			2015-08-14 07:12:14		-27	1869	<blockquote>\n Apologize :- </stron...	5013940.0	...	2018-12-19 11:25:37	
2	15602391	1			2013-03-24 18:35:08		-23	466	I am trying to check if text in editText is...		...	2015-09-25 09:22:16	
3	53386540	1	53386701.0		2018-11-20 05:06:04		-22	1081	What is the shortcut to cr...	8953425.0	...	2019-06-29 03:57:33	
4	43362754	1	43362872.0		2017-04-12 07:12:47		-20	285	My app crashed when I choose time using a T...	6822102.0	...	2017-04-12 07:41:33	
...	...	...	...	...	...	...	...	...	...	...	...	...	
5	38085180	1	38085541.0		2016-06-28 19:54:36		419	545899	I have an input: \n<pre class="lang-html...	1090463.0	...	2021-10-10 09:06:08	
6	42749973	1	42753045.0		2017-03-12 16:31:41		411	142566	I'm starting out a new vue.js project so I ...	1934903.0	...	2021-06-18 10:22:37	

سپس از طریق کد زیر از طریق doc یه object میسازیم .

سپس با iterrow میایم رکورد به رکورد اطلاعات و میخونیم و یه doc میسازیم برای هر سربرگ و اطلاعات آنها.

سپس اطلاعات id , row و میریزیم تو index ها

سپس با map(pd.read_csv, files_list) میایم لیست و میخونیم و بعد از طریق تابع read_csv بریزیم تو data

سپس میایم یه syntax list در میاریم از مواردی که تو برنامه نویسی استفاده میشه و میایم کوئری میزنیم .

```
for index, row in data.iterrows():
    doc = {

        "PostTypeId" : row['PostTypeId'],
        "AcceptedAnswerId" : row['AcceptedAnswerId'],
        "ParentId" : row['ParentId'],
        "CreationDate" : row['CreationDate'],
        "DeletionDate" : row['DeletionDate'],
        "Score" : row['Score'],
        "ViewCount" : row['ViewCount'],
        "Body" : row['Body'],
        "LastEditDate" : row['LastEditDate'],
        "LastActivityDate" : row['LastActivityDate'],
        "Title" : row['Title'],
        "Tags" : row['Tags'],
        "AnswerCount" : row['AnswerCount'],
        "CommentCount" : row['CommentCount'],
        "FavoriteCount" : row['FavoriteCount'],
        "ClosedDate" : row['ClosedDate'],
        "CommunityOwnedDate" : row['CommunityOwnedDate'],
        "ContentLicense" : row['ContentLicense'],
    }
    res = client.index(index="question",id=row['Id'],body=doc)
    print(res['result'])
```

سپس از طریق کد زیر فایل های جدا جدا questionرا index میکینیم داخل الستیک سرچ:

```
dir_path = r'C:\\App\\Data Files\\Answer-Copy\\'

files_list=os.listdir(dir_path)
files_list=['C:\\App\\Data Files\\Answer-Copy\\'+x for x in files_list]

data = pd.concat(map(pd.read_csv, files_list))
data = data.fillna("")
data
```

سپس خروجی زیر را میدهد:

	Id	PostId	Score	Text	CreationDate	UserDisplayName	UserId	ContentLicense
0	98155139	55729619	0	I have an idea: Is it possible if we can run a...	2019-04-18 03:57:01		5677024.0	CC BY-SA 4.0
1	98155688	55739184	0	add your code so that we can understand your p...	2019-04-18 04:38:57		7598082.0	CC BY-SA 4.0
2	98155383	55729619	0	@kiranBiradar Please explain a little more abo...	2019-04-18 04:15:50		5677024.0	CC BY-SA 4.0
3	31322967	20874296	0	I think he does need to use rawQuery here, but...	2014-01-01 23:28:26		95462.0	CC BY-SA 3.0
4	98155040	55724477	0	@davidgyoung You are right, thanks a lot your ...	2019-04-18 03:48:20		8020399.0	CC BY-SA 4.0
...	...	...	...	...	...	...	...	...
5	79940085	35914069	27	How can this be accomplished without vue-router?	2017-09-29 16:07:05		2031033.0	CC BY-SA 3.0
6	68066526	40408096	25	This is the most confusing thing about Vue in ...	2016-11-03 17:56:55		192729.0	CC BY-SA 3.0
7	109123609	35969974	24	hey you, google searcher! if you're reading th...	2020-05-09 05:29:21		1907888.0	CC BY-SA 4.0
8	90915365	50865828	21	@Jeff are you a politician, if not you should ...	2018-08-23 09:50:13		501827.0	CC BY-SA 4.0
9	78673500	45856929	20	You could do something like `next({ path: '/lo...	2017-08-24 09:28:19		1230302.0	CC BY-SA 3.0

همچنین از طریق کد زیر فایل های جدا جدا answer را index میکینیم داخل الستیک سرچ:

```
for index, row in data.iterrows():
    doc = {

        "PostId" : row['PostId'],
        "Score" : row['Score'],
        "Text" : row['Text'],
        "CreationDate" : row['CreationDate'],
        "UserDisplayName" : row['UserDisplayName'],
        "UserId" : row['UserId'],
        "ContentLicense" : row['ContentLicense'],
    }
    res = client.index(index="answer",id=row['Id'],body=doc)
    print(res['result'])
```

سپس برای کوئری هایی که ابتدا جواب های دارای کد و سایت و سپس فقط کد یا سایت و سپس هیچکدام را نشان دهد از کد زیر استفاده میشود که یک نمونه زده شد و خروجی زیر را داد:

Result[hits] [hits]رکورد هایی که پیدا شده را برمیگرداند.

بعد میندازیمش تو حلقه for،id هر رکوردی که به دست اورده و میریزه تو num

```
def search_text(text):

    syntax_list = [']','[',')','(','{','}','if', 'while', 'for', 'class', 'public', 'private', '=']

    results = client.search(index='question',body={"query": {"match_phrase":{"Body":text}}})
    all_hits = results['hits']['hits']

    for num, doc in enumerate(all_hits):
        num = doc["_id"]
        results2 = client.search(index='answer', body={"query": {"match_phrase":{"PostId":num}}})
        if results2['hits']['total']['value'] != 0:
            all_hits2 = results2['hits']['hits']
            for num2, doc2 in enumerate(all_hits2):
                str_text = doc2['_source']['Text']
                urls_list = re.findall(r'(https?://[^\s]+)', str_text)
                syntax_list = ['ok' for i in syntax_list if i in str_text]

                # Step 2
                if len(urls_list) != 0 and len(syntax_list) != 0 :
                    print ("DOC ID:", doc2["_id"], "\n", doc2['_source']['Text'], "\n")
                elif len(urls_list) != 0:
                    print ("DOC ID:", doc2["_id"], "\n", doc2['_source']['Text'], "\n")
                elif len(syntax_list) != 0:
                    print ("DOC ID:", doc2["_id"], "\n", doc2['_source']['Text'], "\n")
                else :
                    print ("DOC ID:", doc2["_id"], "\n", doc2['_source']['Text'], "\n")
```

```
search_text('Vue.js')
```

```
DOC ID: 59490692
 There is not even close to a duplicate. [tag:vue.js] is a framework with a specific logic, different from vanila javascript

DOC ID: 79940085
 How can this be accomplished without vue-router?
```

سوال 3:

در اینجا ابتدا 2 تا فایل tagname , tag ساختیم که یکیش تگای خالیه یکیش ... ,tagname , countهست.id

تو خط totaldata.fillnaبه رشته خالی تبدیل کردم.

بعد csvdata ردیف هایی که قراره بزاریم و تعریف میکنیم.

بعد یه for میزنیم که بیاد رکورد به رکورد دیتا فایل تگ و خوندیم اول اسمشو خونیدم و تگ والدی که تعریف کرئیم و تو for بعدی استفاده میکنیم.

حال اسمی که داریم و تو total tag سرچش میکنیم و id تک رکود و برمیگردونیم

توbaseurl از stackexchange.com یه api هست که اگه هر تگی و بهش بدیم تگ های مرتبط و باهاش و نشون میده

اومدیم ریکوست دادیم و json دریافت کردیم.

برای tagnameاصلی id , name و در میاریم

و بعد از طریق id , name parent تمام related tag هاشو در میاریم.

Time.sleep به خاطر اینه که تند تند درخواست ندیم اینو گذاشتیم که هر یک دقیقه یک بار درخواست بده که ip , block نکنه.

```
import requests
import json
import csv
import pandas as pd
import time
import numpy as np

data = pd.read_csv('TagName.csv')
totalData = pd.read_csv('Tags.csv')
totalData.fillna("")

with open('Result.csv', 'w', encoding='utf-8-sig', newline='') as file:
    csvdata=[]
    writer = csv.writer(file)

    csvdata.append('Parent Tag')
    csvdata.append('Parent ID')
    csvdata.append('Child Tag')
    csvdata.append('Child ID')
    writer.writerow(csvdata)
    csvdata=[]
    counter = 0
    for index, row in data.iterrows():

        parentTagName = row['TagName']
        parentTagId = 0
        parentTagData = totalData[totalData["TagName"] == parentTagName][['Id']]

        for index , row in parentTagData.iterrows():
            parentTagId = row['Id']

        baseURL = 'https://api.stackexchange.com/2.3/tags/{}/related?site=stackoverflow'.format(parentTagName)
        r = requests.get(baseURL)
        json = r.json()
        for i in json['items']:
            name = i['name']
            search = totalData[totalData["TagName"] == name][['Id','TagName']]
            for index , row in search.iterrows():
                csvdata.append(parentTagName)
                csvdata.append(parentTagId)
                csvdata.append(row['TagName'])
                csvdata.append(row['Id'])
                writer.writerow(csvdata)
                csvdata = []
                print('Success...')

        print(++counter, 'Setp ----------------------------------------------')
        time.sleep(60)
```