ADS509_CarrA_Final_Project_02_MySQL_v1

June 24, 2023

1 509 Final Project

This notebook queries the URLs from the MySQL table, scrapes the entire URL HTML contents, then inserts it into a pandas dataframe. The scraped data is then persisted in two ways: 1) The HTML content is written it back to another column in the MySQL table; 2) A copy of the full DF is written to a CSV file for further processing (i.e., processing).

1.1 Globally import libraries

[1]:
```
import numpy as np
import pandas as pd
import pymysql as mysql
import matplotlib.pyplot as plt
import os
import shutil
import re
import logging
import time
import zipfile
import requests
from bs4 import BeautifulSoup
import datetime as dt
import re
import regex as rex
from collections import defaultdict, Counter
import random
#import mysql.connector

# Set pandas global options
pd.options.display.max_rows = 17
```

1.2 Initiate MySQL connection

[2]:
```
'''Set local environment variables to hide user name & password citation:
https://www.geeksforgeeks.org/how-to-hide-sensitive-credentials-using-python/
'''
user_name = os.environ['MySQLUSRAC']
user_pass = os.environ['MySQLPWDAC']
```

```
# Instantiate connection
db_conn = mysql.connect(host='localhost',
                        port=int(3306),
                        user=user_name,
                        passwd=user_pass,
                        db='509_final_proj')

# Create a cursor object
cursor = db_conn.cursor()
```

[3]:
```
tbl_names = pd.read_sql('SHOW TABLES', db_conn)

display(tbl_names)
print(type(tbl_names))
```

```
C:\Users\acarr\AppData\Local\Temp\ipykernel_26388\4193860975.py:1: UserWarning:
pandas only supports SQLAlchemy connectable (engine/connection) or database
string URI or sqlite3 DBAPI2 connection. Other DBAPI2 objects are not tested.
Please consider using SQLAlchemy.
  tbl_names = pd.read_sql('SHOW TABLES', db_conn)
```

	Tables_in_509_final_proj
0	nar_temp
1	news_articles

```
<class 'pandas.core.frame.DataFrame'>
```

1.2.1 Establish logging policy

[4]:
```
'''Logging citations (see additional code in following code blocks:
OpenAI. (2021). ChatGPT [Computer software]. https://openai.com/;
https://docs.python.org/3/howto/logging.html#logging-basic-example;
https://docs.python.org/3/howto/logging.html#logging-to-a-file;
https://docs.python.org/3/howto
/logging-cookbook.html#using-a-rotating-log-file-handler;
https://docs.python.org/3/howto
/logging-cookbook.html#using-a-timed-rotating-file-handler
'''
# Set up logging
logging.basicConfig(level=logging.INFO,
                    filename='pymysql.log',
                    filemode='a',
                    format='''>>>>>>>>>>>>>>>><<<<<<<<<<<<<<<<\n%(asctime)s -
                    %(levelname)s - %(message)s''')
```

1.2.2 Read URLs from MySQL table to perform web scraping

[5]:
```
nat_tbl_name = 'nar_temp'
nwa_tbl_name = 'news_articles'
```

[6]:
```
'''Connect to MqSQL table in batches citation:
OpenAI. (2021). ChatGPT [Computer software]. https://openai.com/
'''

# Batch size (number of URLs to process at a time)
batch_size = 10000

# Get the total number of URLs in the table
count_query = f"SELECT COUNT(*) FROM {nwa_tbl_name}"
cursor.execute(count_query)
total_urls = cursor.fetchone()[0]
print(f'URL Count: {total_urls}')

# Start timer
start_time = dt.datetime.today()

# Calculate the number of batches required
num_batches = (total_urls // batch_size) + 1

# Process URLs in batches
for batch in range(num_batches):
    offset = batch * batch_size

    # Retrieve URLs from the MySQL table in the current batch
    query = f'''
    SELECT url FROM {nwa_tbl_name}
    WHERE article_text IS NULL
    AND (source_name="CNN"
        OR source_name="The Washington Post"
        OR source_name="Fox News"
        OR source_name="Slate Magazine"
        OR source_name="Vox"
        OR source_name="Breitbart News")
    LIMIT {batch_size}
    OFFSET {offset}
    '''

    print(query)
    cursor.execute(query)
    urls = cursor.fetchall()
    print(f'URL batch size: {len(urls)}')
```

```
    # Iterate over the URLs and scrape their contents
    for url in urls:
        url = url[0]  # Extract the URL from the tuple

        # Make an HTTP request to the URL
        response = requests.get(url)
        time.sleep(5 + 11 * random.random())

        # Check if the request was successful
        if response.status_code == 200:
            # Parse the HTML content using BeautifulSoup
            soup = BeautifulSoup(response.content, 'html.parser')

            # Extract the raw text from the HTML
            #print(soup.prettify())
            #raw_text = soup.get_text()
            raw_text = soup.prettify()


            # Update the MySQL table with the scraped text
            update_query = '''
            UPDATE news_articles SET article_text = %s
            WHERE url = %s
            '''

            print('.', end='')
            #print(update_query)
            cursor.execute(update_query, (raw_text, url))
            db_conn.commit()
        else:
            print(f'Response: {response.status_code}')

# End timer script
end_time = dt.datetime.today()
time_elapse = end_time - start_time
print(f'Start Time = {start_time}')
print(f'End Time = {end_time}')
print(f'Elapsed Time = {time_elapse}')
```

```
URL Count: 6283
```

```
    SELECT url FROM news_articles
    WHERE article_text IS NULL
    AND (source_name="CNN"
        OR source_name="The Washington Post"
        OR source_name="Fox News"
        OR source_name="Slate Magazine"
        OR source_name="Vox"
```

```
        OR source_name="Breitbart News")
    LIMIT 10000
    OFFSET 0

URL batch size: 41
Response: 404
Response: 404
…Start Time = 2023-06-18 00:56:01.517790
End Time = 2023-06-18 01:15:26.593576
Elapsed Time = 0:19:25.075786
```

1.2.3 Send MySQL records to CSV

[7]:
```
slct_tbl_full_df01 = pd.read_sql(
                                 '''
                                 SELECT * FROM news_articles
                                 WHERE article_text IS NOT NULL
                                     AND (source_name="CNN"
                                     OR source_name="The Washington Post"
                                     OR source_name="Fox News"
                                     OR source_name="Slate Magazine"
                                     OR source_name="Vox"
                                     OR source_name="Breitbart News")
                                 ''',
                                 db_conn)
```

```
C:\Users\acarr\AppData\Local\Temp\ipykernel_26388\1187134288.py:1: UserWarning:
pandas only supports SQLAlchemy connectable (engine/connection) or database
string URI or sqlite3 DBAPI2 connection. Other DBAPI2 objects are not tested.
Please consider using SQLAlchemy.
  slct_tbl_full_df01 = pd.read_sql(
```

[8]:
```
'''Dir nav citation:
https://softhints.com/python-change-directory-parent/
'''
curr_dir = os.path.abspath(os.curdir)
print(curr_dir)
os.chdir("..")
up1_dir = os.path.abspath(os.curdir)
print(up1_dir)
```

```
C:\Users\acarr\Documents\GitHub\ADS509_Final_project\deliverables
C:\Users\acarr\Documents\GitHub\ADS509_Final_project
```

[9]:
```
# change `data_location` to the location of the folder on your machine.
data_location = 'data_restricted'

file_name = 'data_raw_amc.csv'
```

```
file_path = os.path.join(up1_dir, data_location, file_name)

print(f'CSV file path: {file_path}')
```

```
CSV file path: C:\Users\acarr\Documents\GitHub\ADS509_Final_project\data_restric
ted\data_raw_amc.csv
```

[10]:
```
slct_tbl_full_df01.to_csv(file_path, index=False)
```

[11]:
```
print(type(slct_tbl_full_df01))
display(slct_tbl_full_df01.head(11))
#display(slct_tbl_full_df01['article_text'][0])
```

```
<class 'pandas.core.frame.DataFrame'>
```

	text_id	source_name	author	title	url	publish_date
0	1	CNN	Clare Foran,Nicky Robertson	Senate races to avert default but vote timing …	https://www.cnn.com/2023/06/01/politics/senate…	2023-06-01T09:00:40Z
1	2	Fox News	Paul Steinhauser	First on Fox: Pro-Tim Scott super PAC launches…	https://www.foxnews.com/politics/pro-scott-sup…	2023-06-01T16:12:56Z
2	3	Fox News	Greg Wehner	Pennsylvania bus driver allegedly used duct ta…	https://www.foxnews.com/us/pennsylvania-bus-dr…	2023-05-31T00:21:20Z
3	4	Fox News	Michael Ruiz	Bob Lee murder: Cash App founder seen with sus…	https://www.foxnews.com/us/bob-lee-murder-cash…	2023-05-31T20:30:12Z
4	5	Fox News	Brooke Singman	Senate GOP demands answers on security clearan…	https://www.foxnews.com/politics/senate-gop-de…	2023-05-31T22:11:38Z
5	6	Fox News	Peter Kasperowicz	Will AI ever be smart enough to decipher feder…	https://www.foxnews.com/politics/ai-smart-enou…	2023-06-01T06:00:30Z
6	7	Fox News	Fox News Staff	SEAN HANNITY: Here's what you need to know abo…		
7	8	Fox News	Melissa Rudy	Ozempic, Wegovy and pregnancy risk: What you n…		
8	9	Fox News	Associated Press	NATO ramps up pressure on Turkey to drop objec…		
9	10	Fox News	Fox News Staff	Mike Lee goes off on Biden-McCarthy debt ceili…		
10	11	Fox News	Associated Press	Indiana police officer, suspect hospitalized f…		

```
6   https://www.foxnews.com/media/sean-hannity-her…  2023-05-31T02:53:55Z
7   https://www.foxnews.com/health/ozempic-wegovy-…  2023-06-01T15:58:10Z
8   https://www.foxnews.com/world/nato-ramps-press…  2023-06-01T16:54:03Z
9   https://www.foxnews.com/media/mike-lee-goes-bi…  2023-06-01T17:00:26Z
10  https://www.foxnews.com/us/southern-indiana-po…  2023-05-31T12:39:31Z

                                          article_text content
0   <!DOCTYPE html>\n<html data-layout-uri="cms.cn…    None
1   <!DOCTYPE html>\n<html data-n-head="%7B%22lang…    None
2   <!DOCTYPE html>\n<html data-n-head="%7B%22lang…    None
3   <!DOCTYPE html>\n<html data-n-head="%7B%22lang…    None
4   <!DOCTYPE html>\n<html data-n-head="%7B%22lang…    None
5   <!DOCTYPE html>\n<html data-n-head="%7B%22lang…    None
6   <!DOCTYPE html>\n<html data-n-head="%7B%22lang…    None
7   <!DOCTYPE html>\n<html data-n-head="%7B%22lang…    None
8   <!DOCTYPE html>\n<html data-n-head="%7B%22lang…    None
9   <!DOCTYPE html>\n<html data-n-head="%7B%22lang…    None
10  <!DOCTYPE html>\n<html data-n-head="%7B%22lang…    None
```

1.2.4 Commit changes and close cursor and connection instances

[12]:
```
# Commit the changes to the database
db_conn.commit()

# Close the cursor and database connection
cursor.close()
db_conn.close()
```