Web scraping & Text mining based Information Retrieval for Student Admission Assistance

Bhavna Bharath
Department of Computer Science
Stevens Institute of Technology
bbharath@stevens.edu

Barun Roy
Department of Computer Science
Stevens Institute of Technology
broy2@stevens.edu

Sushmitha Anjanappa
Department of Computer Science
Stevens Institute of Technology
sanjanap@stevens.edu

***Abstract*—This paper covers a novel approach for extraction of admission related information for prospective students around the globe using a text mining based web crawling and web scrapping technique. This is particularly beneficial which can be used to obtain most pertinent details required for making admission and application decisions by the student community. Our system will display relevant information related to location, university names, scores, ratings and price from a university finder website in order to make decisions based on the above-mentioned parameters. Users can extract data by selecting any country and search for colleges by inputting values according to their preference. The system will restructure the data required by the user in a readable format and display it in the form of tables and graphs for easy analysis and decision making. It makes it very easy for students who are in the lookout for colleges for higher studies and can fetch data very quickly with respect to price, ranking and location factors. This paper also covers how transforming data in the form of Graphical plots with respect to university cost/fee and ranking is so convenient for decision making.**

Keywords—Web scrapping, web crawling, text mining

I. INTRODUCTION

Web scrapping is fundamentally an automated interactive scheme for website and some other online resources to access data. This is particularly useful when you want to mine information and keep it in an external archive for review. The system uses Beautiful soup which is a tool that provides a powerful interface for scraping and parsing data from the web. This in turn speeds up the data collection process. While browsing, it is extremely a challenging task to go through the entire website for information by means of clicking and scrolling and a lot of time is wasted in this process. Though there are multiple advantages, there are some significant challenges when it comes to scraping the data from a variety of websites due to its complex structure since each website is unique in its own way. One more major challenge is the dynamicity of websites since they are in constant development stage and it might be difficult to scrap content and navigate to important information. Web scrapping technique is similar to text mining where it has broader set of applications associated to data analytics, e-commerce etc. The scraped content can be used for database development or any research purposes.

A web crawler's core requirement is to automate gathering useful information that are inaccessible pieces of data from websites and this project essentially deals with a university finder website called Collegeduniya. Web scraping technology acts as an interface between the websites and the resulting structured information. This method of obtaining data makes it consistent and easy to handle. The proposed solution highly augments the quality of data applicability and is built for text mining, web scrapping combined with web crawling to provide solutions that examines the website in place.

This project essentially focuses on delivering clean and organized data consisting of University name, rating and score requirements for application process. It fetches only these data from the Collegeduniya website and aggregates all relevant data into a table format with required fields. This project revolves around exploring the website, inspecting it, deciphering URLs, scraping the HTML content from a webpage (can be static or dynamic), parsing the HTML tags using Beautiful soup python library which helps in parsing structured data and extracting the text content from HTML elements & attributes. The main motivation of this project is to perform advanced parsing, navigation and extraction of admission related information from a university finder website that can aid student community (predominantly who are not aware of which universities to apply for). The system assists students to get past the confusion that is in place when it comes to browsing a huge website for application requirements. The prime focus of the project is to generate structured data from unstructured data available on the Internet. The proposed methodology obtains web -based information and integrates it to a specific repository (.csv file) and also performs graphical analysis on the obtained data. There are multiple ways to dynamically scrap data namely pattern matching, web mining, Parsing through HTML tags and many more. The are multiple fields where web scrapping is used such as Banking sector, Insurance, Finance, IT sector, Marketing and Education.

The following sections cover the implementation of the proposed methodology and scope for future work.

II. ANALYSIS & CODE

Coding and testing is done using Python language in Jupyter notebook from Anaconda navigator. The software is open source and free to use.

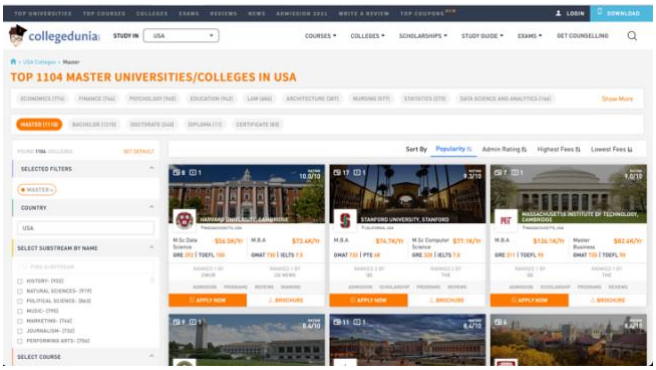
```
In[]: from selenium.webdriver.common.action_chains
import ActionChains
from selenium.webdriver.support import
expected_conditions as EC
from selenium.common.exceptions import
NoAlertPresentException
from selenium.common.exceptions import
```

```
NoSuchElementException
from webdrivermanager.chrome import
ChromeDriverManager
from selenium.webdriver.support.ui import
WebDriverWait
from selenium.common.exceptions
import TimeoutException
from selenium.webdriver.support.ui import Select
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.common.by import By
from bs4 import BeautifulSoup as bs
import matplotlib.ticker as ticker
from urllib.request import urlopen
import matplotlib.dates as mdates
import matplotlib.pyplot as plt
from selenium import webdriver
from bs4 import BeautifulSoup
from datetime import datetime
from selenium import webdriver
import seaborn as sns
import pandas as pd
import numpy as np
import selenium
import requests
import unittest
import time
import re
import sys

In[]: def scroll(driver, timeout):
scroll_pause_time = timeout
# Get scroll height
last_height = driver.execute_script("return
document.body.scrollHeight")
while True:
# Scroll down to bottom
driver.execute_script("window.scrollTo(0,
document.body.scrollHeight);")
# Wait to load page
time.sleep(scroll_pause_time)
# Calculate new scroll height and compare with last scroll
height
new_height = driver.execute_script("return
document.body.scrollHeight")
if new_height == last_height:
# If heights are the same it will exit the function
break
last_height = new_height
def all_links(url):
headers = {"User-Agent":"Mozilla/5.0 (Windows NT 10.0;
Win64; x64; rv:66.0) Gecko/20100101 Firefox/66.0",
"Accept-Encoding":"gzip, deflate",
"Accept":"text/html,application/xhtml+xml,application/xml;
q=0.9,*/*;q=0.8", "DNT":"1","Connection":"close",
"Upgrade-Insecure-Requests":"1"}

# Setup the driver. This one uses chrome with some
options and a path to the chromedriver
driver = webdriver.Chrome()
# implicitly_wait tells the driver to wait before throwing
an exception
driver.implicitly_wait(5)
# driver.get(url) opens the page
driver.get('https://collegedunia.com/germany-colleges')
# This starts the scrolling by passing the driver and a timeout
scroll(driver, 5)
# Once scroll returns bs4 parsers the page_source
soup = BeautifulSoup(driver.page_source, 'lxml')
# Then we close the driver as soup_a is storing the page
source
driver.close()
alls = []
for d in soup.findAll('div', attrs={'class':'jsx-1879893061
listing-block text-uppercase bg-white position-relative'}):

name = d.find('h3', attrs={'class':'jsx-1879893061 text-white
font-weight-bold text-md m-0'})
location = d.find('span', attrs={'class':'jsx-1879893061 mr-
1'})
rating = d.find('span', attrs={'class':'jsx-1879893061 rating-
text text-white font-weight-bold text-base d-block text-
right'})
scores = d.find('span', attrs={'class':'jsx-1879893061 d-
block'})
price = d.find('span', attrs={'class':'jsx-1879893061 d-flex
justify-content-between'})
all1=[]
                if name is not None:
                    #print(n[0]['alt'])
                    all1.append(name.text)
                else:
                    all1.append("N/A")
                if location is not None:
                    all1.append(location.text)
                else:
                    all1.append('N/A')
                if rating is not None:
                    #print(rating.text)
                    all1.append(rating.text)
                else:
                    all1.append('N/A')
                if scores is not None:
                    #print(price.text)
                    all1.append(scores.text)
                else:
                    all1.append('N/A')
                if price is not None:
                    #print(price.text)
                    all1.append(price.text)
                else:
                    all1.append('N/A')
                alls.append(all1)
    return alls

    In[]: results = []
    for i in range(1):
    results.append(all_links(i))
    flatten = lambda l: [item for sublist in l for item in
    sublist]
    df =
    pd.DataFrame(flatten(results),columns=['UNIVERSITY
    NAME','LOCATION','RATING','SCORES','PRICE'])
    #cleaning data for ML and Data Engineering.
    df["UNIVERSITY NAME"] = df["UNIVERSITY
```

```
NAME"].str.replace(',', ")
df['UNIVERSITY NAME'] =
pd.to_numeric(df['UNIVERSITY NAME'],
errors='ignore')
df["LOCATION"] = df["LOCATION"].str.replace(',', ")
df["LOCATION"] =
df["LOCATION"].str.replace('germany', ")
df['RATING'] = df['RATING'].str.replace('/', ")
df['RATING'] = df['RATING'].str.replace('10', ")
df['PRICE'] = df['PRICE'].str.extract('([0-9][,.]*[0-9]*)')
#df['PRICE'] = df['PRICE'].apply(lambda x:
find_number(x))
df[['GREGMAT','IELTSTOEFL']] =
df.SCORES.str.split("|",expand=True)
df.to_csv('germany.csv', index=False, encoding='utf-8')
```

In[]: df = pd.read_csv("germany.csv")
df.shape

In[]: df.head(10)



Out[7]:

	UNIVERSITY NAME	LOCATION	RATING	SCORES	PRICE	GREGMAT	IELTSTOEFL
0	University of Freiburg Freiburg	Baden-Wurttemberg	6.2	TOEFL 92	2.9	TOEFL 92	NaN
1	University of Wuerzburg Wurzburg	Bavaria	6.2	NaN	NaN	NaN	NaN
2	Humboldt University of Berlin Berlin	Berlin	6.2	NaN	NaN	NaN	NaN
3	Technical University Munich Munich	Bavaria	6.2	GRE 600 | TOEFL 88	144.0	GRE 600	TOEFL 88
4	Ludwig Maximilian University of Munich Munich	Bavaria	6.2	IELTS 5.5	287.0	IELTS 5.5	NaN
5	Free University of Berlin Berlin	Berlin	6.1	IELTS 5.0	NaN	IELTS 5.0	NaN
6	University of Bonn Bonn	North Rhine-Westphalia	5.9	IELTS 6	666.0	IELTS 6	NaN
7	RWTH Aachen University Aachen	North Rhine-Westphalia	5.9	IELTS 6.5	17.0	IELTS 6.5	NaN
8	Karlsruhe Institute of Technology Karlsruhe	Baden-Wurttemberg	5.7	NaN	3.3	NaN	NaN
9	Heidelberg University Heidelberg	Baden-Wurttemberg	5.7	NaN	382.0	NaN	NaN

In[]: df.dtypes

```
Out[8]: UNIVERSITY NAME      object
        LOCATION             object
        RATING              float64
        SCORES               object
        PRICE               float64
        GREGMAT              object
        IELTSTOEFL           object
        dtype: object
```

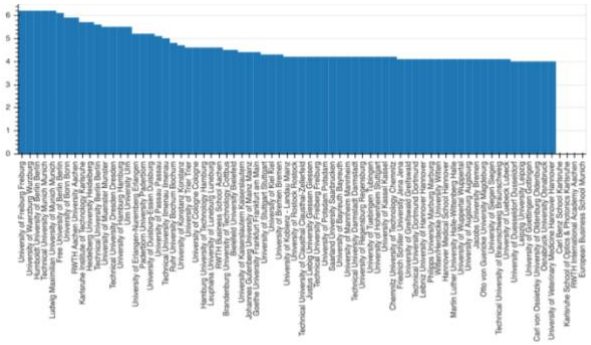
In[]: data = df.sort_values(["RATING"], axis=0,
ascending=False)[:75]
Data

Out[9]:

	UNIVERSITY NAME	LOCATION	RATING	SCORES	PRICE	GREGMAT	IELTSTOEFL
0	University of Freiburg Freiburg	Baden-Wurttemberg	6.2	TOEFL 92	2.9	TOEFL 92	NaN
1	University of Wuerzburg Wurzburg	Bavaria	6.2	NaN	NaN	NaN	NaN
2	Humboldt University of Berlin Berlin	Berlin	6.2	NaN	NaN	NaN	NaN
3	Technical University Munich Munich	Bavaria	6.2	GRE 600 | TOEFL 88	144.0	GRE 600	TOEFL 88
4	Ludwig Maximilian University of Munich Munich	Bavaria	6.2	IELTS 5.5	287.0	IELTS 5.5	NaN
...	...	...	...	...	...	...	...
70	University of Veterinary Medicine Hannover Han...	Lower Saxony	4.0	NaN	NaN	NaN	NaN
71	Carl Benz School Karlsruhe	Baden-Wurttemberg	NaN	TOEFL 88	19.0	TOEFL 88	NaN
72	Karlsruhe School of Optics & Photonics Karlsruhe	Baden-Wurttemberg	NaN	IELTS 6.5	3.3	IELTS 6.5	NaN
73	RWTH International Academy Aachen	North Rhine-Westphalia	NaN	IELTS 5.5	10.0	IELTS 5.5	NaN
74	European Business School Munich	Bavaria	NaN	Estd. Year	13.0	Estd. Year	NaN

In[]: show(p)

```
In[]: from bokeh.models import ColumnDataSource
from bokeh.transform import dodge
import math
from bokeh.io import curdoc
curdoc().clear()
from bokeh.io import push_notebook, show,
output_notebook
from bokeh.layouts import row
from bokeh.plotting import figure
from bokeh.transform import factor_cmap
from bokeh.models import Legend
output_notebook()

In[]: p = figure(x_range=data.iloc[:,0], plot_width=800,
plot_
height=550,
title=" University Ranking based plot for analysis ",
toolbar_
location=None, tools="")
p.vbar(x=data.iloc[:,0], top=data.iloc[:,2], width=0.9)
p.xgrid.grid_line_color = None
p.y_range.start = 0
p.xaxis.major_label_orientation = math.pi/2
```



In[]: data = df.sort_values(["PRICE"], axis=0,
ascending=False)[:75]
Data

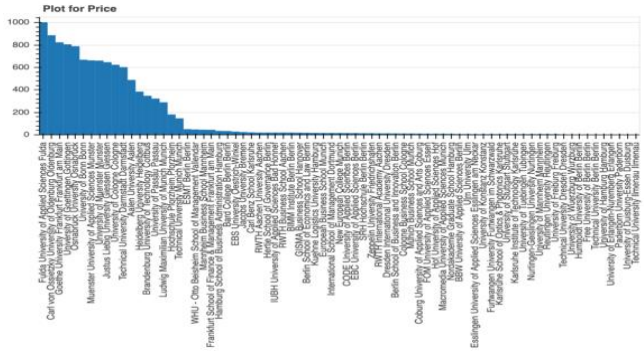
Out[13]:

	UNIVERSITY NAME	LOCATION	RATING	SCORES	PRICE	GREGMAT	IELTSTOEFL
116	Fulda University of Applied Sciences Fulda	Hesse	NaN	TOEFL 85	999.0	TOEFL 85	NaN
66	Carl von Ossietzky University of Oldenburg Old...	Lower Saxony	4.0	TOEFL 43	884.0	TOEFL 43	NaN
31	Goethe University Frankfurt Frankfurt am Main	Hesse	4.4	TOEFL 100	821.0	TOEFL 100	NaN
67	University of Goettingen Gottingen	Lower Saxony	4.0	TOEFL 81	803.0	TOEFL 81	NaN
69	Osnabruck University Osnabruck	Lower Saxony	4.0	NaN	786.0	NaN	NaN
...	...	...	...	...	...	...	...
13	University of Hamburg Hamburg	Hamburg	5.5	TOEFL 92	NaN	TOEFL 92	NaN
15	University of Erlangen-Nuremberg Erlangen	Bavaria	5.2	TOEFL 95	NaN	TOEFL 95	NaN
16	Paderborn University Paderborn	North Rhine-Westphalia	5.2	IELTS 6.0	NaN	IELTS 6.0	NaN
17	University of Duisburg-Essen Duisburg	North Rhine-Westphalia	5.2	NaN	NaN	NaN	NaN
19	Technical University Ilmenau Ilmenau	Thuringia	5.0	TOEFL 79	NaN	TOEFL 79	NaN

75 rows × 7 columns

In[]: p = figure(x_range=data.iloc[:,0], plot_width=800, plot_height=550, title="Price based plot for analysis", toolbar_location=None, tools="")

p.vbar(x=data.iloc[:,0], top=data.iloc[:,4], width=0.9)
p.xgrid.grid_line_color = None
p.y_range.start = 0
p.xaxis.major_label_orientation = math.pi/2



III. CONCLUSION & FUTURE WORK

Firstly, extracting data through web scraping and web crawling is an evolving technology in the IT arena. The traditional or manual way of extracting data makes it very difficult for developers like us to process and deliver data. Scraping enables automation and speed with which data can be manipulated. This project will definitely benefit students around the globe in selecting Universities according to their location preference, GRE/GMAT scores, ranking of the college and expenses that they have to bear. It gives a clear picture on all required contents among the huge dataset available on the Internet. It also makes networking very easy and accessible. There is scope for improvement and the system can also be updated to display college scholarship, funding and counseling related information. The most trending research of a particular University can also be obtained. There are multiple algorithmic implementations and time complexities involved in web scraping and text mining which can be explored for application-oriented research in the future since cleaning the unstructured data is a huge challenge. It is also important to remember that since web scraping involves data produced by others, there are certain ethics for performing web scraping and this project follows all web scraping code of conduct and respects privacy of data produced by others and eliminates misuse.

IV. REFERENCE

[1] Giulio Barcaroli, Alessandra Nurra, and Marco Scarno, "*Use of web scraping and text mining techniques in the Istat survey on Information and Communication Technology in enterprises*", European Conference on Quality in Official Statistics,vol. June 2014

[2] Rushabh A Patel, Mansi Patel, "*A survey on Information retrieval from web using web scraping*", IJIRT vol. 1 Issue 6,2014