import regex as re

import requests

from bs4 import BeautifulSoup

from selenium.webdriver.common.by import By

from selenium import webdriver

from selenium.webdriver.chrome.service import Service

from webdriver\_manager.chrome import ChromeDriverManager

from selenium.webdriver.common.keys import Keys as KeysBrowser

import pandas as pd

import time

from matplotlib import pyplot as plt

driver = webdriver.Chrome(service=Service(ChromeDriverManager().install()))

driver.get("https://www.bailii.org/form/search\_cases.html")

# Enter search query for "Section 172"

search\_box = driver.find\_element(By.NAME, "phrase")

search\_box.send\_keys("Companies Act 2006")

search\_box.send\_keys(KeysBrowser.RETURN)

citation\_urls = []

while True:

soup = BeautifulSoup(driver.page\_source)

ol = soup.find\_all("ol")[0] # Collect the ol element from the page

if not ol:

print("No ol elements found")

break

for li in ol.find\_all("li"):

citation\_link = li.find("a")

if citation\_link:

citation\_url = "https://www.bailii.org/" + citation\_link["href"]

#print(citation\_url)

citation\_urls.append(citation\_url)

try:

# Check if the "Next 10" button exists

#next\_button = driver.find\_element(By.XPATH, "//input[@value='Next 10 >>>']")

except:

# If the button is not found, break out of the loop

break

next\_button.click() # Click the "Next 10" button

time.sleep(2) # Wait for the next page to load

close the browser

driver.quit()

# Define search terms and patterns

search\_terms = ["Section 172", "Sections 172", "section 172", "sections 172", "ss 172", "SS 172"]

keywords\_pattern = re.compile("|".join(search\_terms))

context\_pattern = re.compile(r"(?s)(.{0,200})(Companies Act 2006)(.{0,200})")

# Collect relevant URLs

relevant\_urls = []

for url in citation\_urls:

response = requests.get(url)

soup = BeautifulSoup(response.content, "html.parser")

text = soup.get\_text()

match = context\_pattern.search(text)

if match and (keywords\_pattern.search(match.group(1)) or keywords\_pattern.search(match.group(3))):

relevant\_urls.append(url)

print(relevant\_urls)

We ran the code on chatgpt and we got this response:

A 403 Forbidden error means that the server is refusing to fulfill the request made by the client (in this case, the Python code) because the client does not have permission to access the requested resource. This error can occur if the server is configured to block certain user agents, or if the server has rate limiting in place to prevent excessive requests.

If you are receiving a 403 Forbidden error when trying to access a URL using the requests library, there are several things you can try:

Check the robots.txt file: The website may have a robots.txt file that specifies which pages are allowed to be crawled by search engines and other web crawlers. You can check if the URL you are trying to access is allowed by the robots.txt file by navigating to https://example.com/robots.txt (replace example.com with the domain name of the website) in your browser.

Use a different user agent: The website may be blocking requests from common user agents used by web crawlers. You can try changing the user agent string in the headers parameter of the requests.get() function to mimic a different user agent, such as a web browser.