Cheating Bot Detector

Fall Term Retrospective

Grein, Cameron Gareth
Department of Electrical Engineering
and Computer Science,
Oregon State University
Corvallis, Oregon, United States
greinc@oregonstate.edu

Yidong Lin
Department of Electrical Engineering
and Computer Science,
Oregon State University
Corvallis, Oregon, United States
linyid@oregonstate.edu

Section 1

Due to the continuous improvement of the Internet, learning aid websites such as Chegg and Coursehero have appeared. They claim to be a social learning network. Users can access old test questions, assignments, answers, notes, etc. through these sites. Of course, these are not free, users need to pay a certain amount to continue to access the content. However, you can still browse the content by publishing the materials, and the 40 documents will provide users with 1 month of free browsing. This has led to the leakage of teaching content in many schools. The reason why our product is beneficial that not only the content of Oregon State University's courses has been compromised. On these websites you can find almost all university courses, tests, and answers. Our current work is only for Oregon State University, if the product can achieve the desired effect. The beneficiaries of the product can be more schools. In relation to the scope of the assignment, we should be able to make a basic scraper that can identify websites that contain the course information. Scraping things such as scrape should be quite achievable in the time span that we are provided. Doing things such as scraping images might be a little too ambitious for the time that we have. There also is the question of if we want this to be used as a detector to manually inform that the content needs to be taken down, or are we planning on implementing an automatic system. An automatic system may be out of the scope of the time that we have to complete this assignment.

Section 2

We have made some progress on the implementation of our project. We have completed a python script to search the web for our project. We have not created the user interface yet. The scope of our project seems doable in the timeframe that we have. We should be able to implement all of the features we listed, and what we have also looks good. In terms of scope the current scope of our project seems fine, maybe if we can implement all that we stated in our sprint 1 we could add some more features. Where we are heading in terms of the next sprint, is to finish the basic functionality, and start working on the user interface. We will be working on the project during winter break, hopefully to get more work done. Overall, I think we are doing a good job progression wise.

Section 3

The current problem is that Google will limit the frequency of access. The solution is to set the access delay.

Section 4

spider.py

```python
from requests_html import HTMLSession
from parsel import Selector
import pandas as pd
import time


class Crawler():
    def __init__(self, delay):
        self.session = HTMLSession()
        self.keyword_list = self.get_keywords()
        self.base_url = 'https://www.google.com.hk/'
        self.data_list = []
        self.delay = delay

    def get_keywords(self):
        with open('keywords.txt', 'r', encoding='utf-8-sig') as f:
            keywords = [i.strip() for i in f.readlines() if i.strip()]
        return keywords

    def search(self, url, keyword, page):
        time.sleep(self.delay)
        print(f'>>>>>>Processing keyword <{keyword}> {page}:{url}')
        r = self.session.get(url)
        # with open('test.html','w',encoding='utf-8-sig') as f:
        #     f.write(r.text)
        for i in Selector(r.text).xpath('//*[@id="rso"]/div[@class="g"]'):
            item = {}
            item['keyword'] = keyword
            item['page'] = page
            item['title'] = i.xpath('./div/div/a/h3/span/text()').get('').strip()
            item['desc'] = i.xpath('string(./div/div[2]/div/span)').get('').replace('\n', '').replace('\t', '').replace(
                '\r', '').replace(' ', '').replace('\xa0', '').strip()
            item['url'] = i.xpath('./div/div[2]/div/div/div/a[1]/@href').get('').strip()
            if not item['url']:
                item['url'] = i.xpath('./div/div[1]/a/@href').get('').strip()
            print(item)
            self.data_list.append(item)
```

```
        next_page_url = r.html.xpath('//a[@id="pnnext"]/@href')
        if next_page_url:
            next_page_url = self.base_url + next_page_url[0]
            self.search(next_page_url, keyword, page + 1)

    def save(self):
        df = pd.DataFrame(self.data_list)
        path = f'{str(int(time.time()))}.xlsx'
        with pd.ExcelWriter(path, engine='xlsxwriter', options={'strings_to_urls': False}) as writer:
            df.to_excel(writer, index=False, encoding='utf-8-sig')

    def main(self):
        for keyword in self.keyword_list:
            url = self.base_url + 'search?q=' + keyword
            page = 1
            self.search(url, keyword, page)
        self.save()


if __name__ == '__main__':
    delay = input('>>>>>>Please enter the request delay interval (seconds):\n').strip
    try:
        delay = int(delay)
    except:
        delay = 1
    c = Crawler(delay)
    c.main()
```

- Open the command prompt and run pip install -r requirements.txt
- Confirm the search term: keywords.txt
- Execute python spider.py

Section 5

Regarding some feedback we received when we displayed our project. People were confused at the goals of the project. Since then we have tried to pinpoint the goals of our project more.

There was also some feedback on what language we should use for which parts of our project. Though we feel that the languages and technologies we are using now will suffice.

The last feedback that was given to us was an efficient user interface, and what faculty may want in terms of features for our web crawler.