Advanced Scraping

Adam Kaplan

February 25, 2022

Law/Ethics

Scraping public websites, even against the terms of service, is probably legal. See hiQ Labs, Inc. v. LinkedIn Corp (2019). https://www.eff.org/deeplinks/2019/09/victory-ruling-hiq-v-linkedin-protects-scraping-public-data

But circumventing technical restrictions (passwords, captchas) is probably illegal.

In any case, be nice!

- add a delay between pages

Law/Ethics

Scraping public websites, even against the terms of service, is probably legal. See hiQ Labs, Inc. v. LinkedIn Corp (2019). https://www.eff.org/deeplinks/2019/09/victory-ruling-hiq-v-linkedin-protects-scraping-public-data

But circumventing technical restrictions (passwords, captchas) is probably illegal.

In any case, be nice!

- add a delay between pages
- only use distributed scraping against rich sites

Random headers to avoid blocking

```
desktop_agents = ['Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/54.0.2840.
                  'Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/54.0.2840.
                  'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/54.0.
                  'Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_1) AppleWebKit/602.2.14 (KHTML, like Gecko) Vers
                  'Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/54.0.2840.
                  'Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome

def random_headers():
    return {'User-Agent': choice(desktop_agents),'Accept':'text/html,application/xhtml+xml,application/xml;q=0.

page = requests.get(url, headers=random_headers())
```

Download Youtube videos

Downloading Youtube videos can be important for research reproducibility (see Rich's book).

https://github.com/nficano/pytube

Download Youtube metadata

We probably care just as much about the video metadata

https://github.com/elizariley/youtube_extremism

Downloading a whole bunch of PDFs

```
def download_cable(i, year, collection, outdir):
    docid = str(i)
    collection = "2694"
    year = "1978"
    docname = outdir + year + "_" + docid + ".pdf"
    url = "http://aad.archives.gov/aad/createpdf?rid={0}&dt={1}&dl=2"\
        .format(docid, collection)
    req = urllib2.Request(url)
    r = urllib2.urlopen(req)
    if r.readline():
        f = open(docname, 'wb')
        f.write(r.read())
        f.close()
```

Reverse-engineering API calls

Sometimes you can directly access the raw data that goes into a page's visualization using the "Javascript Console"/network tab.

https://trac.syr.edu/phptools/immigration/arrest/

OCR and Tesseract

If you scrape image PDFs that you want the text from, you'll need to do optical character recognition to convert the image to text

https://github.com/tesseract-ocr/tesseract

Big scrape

- Q: What if you're scraping 10 million stories and you don't want to start over if something breaks?

Big scrape

- Q: What if you're scraping 10 million stories and you don't want to start over if something breaks?
- A: Use queues, databases, and multiple workers

Big scrape

- Q: What if you’re scraping 10 million stories and you don’t want to start over if something breaks?
- A: Use queues, databases, and multiple workers

Big scrape

- Q: What if you're scraping 10 million stories and you don't want to start over if something breaks?
- A: Use queues, databases, and multiple workers

https://github.com/ahalterman/big_scrape (email Andy Halterman at ahalterman0@gmail.com for access)

Newspapers3k

Scraping a bunch of articles from a bunch of sites? This library automatically finds titles, authors, text, etc.

https://newspaper.readthedocs.io/en/latest/

Rendering Javascript

Some page elements are dynamically generated and require Javascript to render.

To scrape, we can use a combination of `PhantomJS` (a windowless browser) and `selenium` (a tool for automating browser actions).