Angel Pablo Hinojosa

Scraping y Obtención de Datos para Big (y no tan Big) Data (Parte I)

Big Data

Lo que significa…

Obtención, gestión y manipulación de grandes volúmenes de datos.

Little Big Data

(Lo mismo, pero menos)

¿De dónde saco la información?

Datos primarios:

Datos de nuestras propias fuentes: usuarios, tráfico, logs, mails, redes, aplicaciones biométricas...

¿De dónde saco la información?

Datos de terceros:

Fuentes abiertas, proveedores de datasets…

(Por ejemplo `http://opendata.ugr.es/`)

...y scraping

Formatos

No todos los formatos son iguales:

- Formatos manipulables y no manipulables
- Formatos abiertos y cerrados

El infierno de los formatos

...y el PDF como ejemplo de todo lo malo

`http://sl.ugr.es/bigdataPDF`

El infierno de los formatos

Ejemplos:

- PRIMER_TRIMESTRE_2015.pdf
-
- presupuesto_ayuntamiento_2015.pdf
-
- presupuesto_parque_ciencias_2015.pdf
-
- tabla.pdf

El infierno de los formatos

...y el PDF como ejemplo de todo lo malo

`http://sl.ugr.es/bigdataPDF`

El infierno de los formatos

Son tus amigos:

- JSON
- CSV

Herramientas online:

Multiconversor:

http://www.cometdocs.com/

(con Limitaciones si no pagas)

Herramientas online:

Extraer tablas de PDF:

`https://pdftables.com/`

(Sólo lee las tablas)

Herramientas online:

PDF a Excell (y otros):

`https://www.pdftoexcelonline.com/`

Lo envía a tu correo

Herramientas online:

OCR de PDF

`http://www.onlineocr.net/`

(una sola pagina)

Herramientas online:

OCR de PDF

`http://free-online-ocr.com/`

(hasta 10 páginas)

NOTA: Ningún OCR es bueno

Lento, complejo, propenso a errores

Requiere supervisión y revisión posterior

Huye del OCR como de la peste

En local

Tabula

`http://tabula.technology/`

(fácil y simple, pero son OCR)

Conversor de formatos

Pandoc

`pandoc.org/`

(no convierte DE PDF)

Para programadores (Python)

pdfminer

`https://euske.github.io/pdfminer/`

pypdfocr

`https://pypi.python.org/pypi/pypdfocr`

Gracias

(Ruegos y preguntas)

Angel Pablo Hinojosa

Scraping y Obtención de Datos para Big (y no tan Big) Data (Parte II)

Orígenes de Datos

Open Data (y transparencia)

Orígenes de Datos

CKAN, Datasets y APIs

`http://opendata.ugr.es`

(Y licencias)

Orígenes de Datos

Catálogo nacional:

`http://datos.gob.es/catalogo`

Orígenes de Datos

Catálogo europeo:

`http://open-data.europa.eu/es/data/`

Orígenes de Datos

Catálogo USA:

http://open-data.europa.eu/es/data/

Orígenes de Datos

Mapa de orígenes:

`http://eip.lcc.uma.es/opendata/`

(poco actualizado)

Web Scraping

"Rascar" datos de Webs

Con sus cuestiones técnicas

Y sus cuestiones Legales

Web Scraping

HTML

La materia de la que están hechas las webs

`http://www.psicobyte.com/html/curso/`

(Tutorial de HTML)

Web Scraping

Import.io

Rudimentario, pero a veces basta

`https://import.io/`

Web Scraping

Scraper (plugin de Chrome)

http://www.psicobyte.com/html/curso/

Web Scraping

Práctica de Scraper

`http://osl.ugr.es`

(...y XPath)

Web Scraping

Caso práctico:

¿Buscamos radares?

http://www.dgt.es/es/el-trafico/control-de-velocidad/granada/

Web Scraping

Usando Google Docs

`https://docs.google.com`

(Google Spreadsheets, concretamente)

Web Scraping

Usando Google Docs (importar feeds)

=IMPORTFEED(“URL”)

Espera ¿Qué es un “feed”?

Web Scraping

Usando Google Docs (importar feeds)

`http://osl.ugr.es/feed/`

Web Scraping

Usando Google Docs (importar HTML -listas-)

=IMPORTHTML(URL,”list”,N)

Web Scraping

Usando Google Docs (importar HTML -listas-)

`http://www.dmoz.org/Computers/Internet/`

Web Scraping

Usando Google Docs (importar HTML -tablas-)

=IMPORTHTML(URL,”table”,N)

Web Scraping

Usando Google Docs (importar HTML -tablas-)

`http://www.dgt.es/es/el-trafico/control-de-velocidad/granada/`

Web Scraping

Usando Google Docs (importar HTML -tablas-)

`http://www.dgt.es/es/el-trafico/control-de-velocidad/granada/`

Web Scraping

Usando Google Docs (importar XML)

=IMPORTXML(URL,”table”,N)

(en realidad, HTML con XPath)

Web Scraping

Usando Google Docs (importar XML)

```
http://osl.ugr.es
```

```
//h2
```

```
//a/@href
```

```
//h2/a/@href
```

Gracias

(Ruegos y preguntas)

Angel Pablo Hinojosa

Scraping y Obtención de Datos para Big (y no tan Big) Data (Parte III)

Trabajar en remoto

Conectando con servidores:

FTP
(con Filezilla)

`cabas.ugr.es`

Trabajar en remoto

(not in Kansas anymore)

Trabajar en remoto

Conectando con servidores:

SSH
(con PuTty)

Shell de UNIX básica:

- ls
- cd

- cat
- more
- less

- touch
- rm
- mv

Shell de UNIX básica:

- uname
- top

- grep
- wc
- less
- sort

- nano
- man

Python (aún más) básico

Hola mundo, abrir ficheros y poco más

```
python miprograma.py
```

Scrapy

Modo interactivo

```
scrapy shell "URL"
```

Scrapy

Modo interactivo:

```
response.xpath('//title')
response.xpath('//title').extract()
response.xpath('//h2')
response.xpath('//h2/text()').extract()
response.xpath('//a')
response.xpath('//a/@href')
```

Scrapy

Modo interactivo:

```
fetch(“URL”)
view(response)
quit
```

Scrapy

Nuestro primer scraper:

`http://doc.scrapy.org/en/latest/intro/tutorial.html`

Scrapy

Nuestro primer scraper:

```
scrapy startproject tutorial
```

Scrapy

Editamos items.py:

```
import scrapy


class DmozItem(scrapy.Item):
    title = scrapy.Field()
    link = scrapy.Field()
    desc = scrapy.Field()
```

Scrapy

Creamos dmoz_spider.py

```
import scrapy

class DmozSpider(scrapy.Spider):
    name = "dmoz"
    allowed_domains = ["dmoz.org"]
    start_urls = [
        "http://www.dmoz.org/Computers/Programming/Languages/Python/Books/",
        "http://www.dmoz.org/Computers/Programming/Languages/Python/Resources/"
    ]

    def parse(self, response):
        filename = response.url.split("/")[-2] + '.html'
        with open(filename, 'wb') as f:
            f.write(response.body)
```

Scrapy

Ejecutamos el scraper:

```
scrapy crawl dmoz
```

Scrapy

Ejecutamos el scraper:

```
scrapy crawl dmoz
```

Scrapy

segunda versión de dmoz_spider.py

```
import scrapy

class DmozSpider(scrapy.Spider):
    name = "dmoz"
    allowed_domains = ["dmoz.org"]
    start_urls = [
        "http://www.dmoz.org/Computers/Programming/Languages/Python/Books/",
        "http://www.dmoz.org/Computers/Programming/Languages/Python/Resources/"
    ]

    def parse(self, response):
        for sel in response.xpath('//ul/li'):
            title = sel.xpath('a/text()').extract()
            link = sel.xpath('a/@href').extract()
            desc = sel.xpath('text()').extract()
            print title, link, desc
```

Scrapy

Ejecutamos (otra vez) el scraper:

```
scrapy crawl dmoz
```

Scrapy

tercera versión de dmoz_spider.py

```
import scrapy, urlparse

from tutorial.items import DmozItem

class DmozSpider(scrapy.Spider):
    name = "dmoz"
    allowed_domains = ["dmoz.org"]
    start_urls = [
        "http://www.dmoz.org/Computers/Programming/Languages/Python/",
    ]

    def parse(self, response):
        for href in response.css("ul.directory.dir-col > li > a::attr('href')"):

            url = urlparse.urljoin(response.url, href.extract())
            yield scrapy.Request(url, callback=self.parse_dir_contents)

    def parse_dir_contents(self, response):
        for sel in response.xpath('//ul/li'):
            item = DmozItem()
            item['title'] = sel.xpath('a/text()').extract()
            item['link'] = sel.xpath('a/@href').extract()
            item['desc'] = sel.xpath('text()').extract()
            yield item
```

Scrapy

Ejecutamos (otra vez) el scraper:

```
scrapy crawl dmoz
```

```
scrapy crawl dmoz > archivo
```

```
scrapy crawl --nologo dmoz
```

Gracias

(Ruegos y preguntas)

http://www.psicobyte.com
@psicobyte
psicobyte@gmail.com