Australian National University

COMP3430 / COMP8430
Data wrangling

Lecture 4: Web scraping and geocoding of data

(Lecturer: Peter Christen)

Lecture outline

- Web scraping
  - Extracting data from the Web
  - Web scraping with Python
- Forward and reverse geocoding
  - Geocoding with Python
- Summary

Extracting data from the Web

- Collection and cleaning of Web data is required in many Web data-driven projects
- Examples: price comparison, product review, real estate listings, stock market, web mashup, web data integration, and research

- Websites contain unstructured and semi-structured data (HTML format)
- Need to extract data into structured format from unstructured data

Ways to extract data from the Web

- APIs (Application Programming Interface)
  - Some websites provide APIs to extract (structured) data
  - Examples: Twitter, Google, Facebook
  - Some APIs are restricted by what data is available to be extracted and how frequent
- Web scraping techniques
  - Cannot always rely on APIs to access Web data
  - Web scraping techniques are developed to transform unstructured data in the Web to structured data

Is Web scraping legal?

- Several case studies around the world
  - *In Feist Publications, Inc. v. Rural Telephone Service Co., the United States Supreme Court decided that scraping and republishing facts, such as telephone listings, is allowed.*
    See: http://caselaw.findlaw.com/us-supreme-court/499/340.html
  - *Then, a similar case in Australia, Telstra Corporation Limited v. Phone Directories Company Pty Ltd, demonstrated that only data with an identifiable author can be copyrighted. See: http://www.austlii.edu.au/au/cases/cth/FCA/2010/44.html*
  - *Also, the European Union case, ofir.dk vs home.dk, concluded that regular crawling is permissible.*
    (Web scraping with Python, Richard Lawson, 2015)

Is Web scraping legal? (2)

- Scraped data used for personal use does not involve legal issues

- However, when re-publishing scraped data the type of data scraped is important:
  - When scraped data contains facts, such as price, location, and contact details, it can be republished
  - Data containing opinions and reviews cannot be republished for copyright reasons

Understanding the target Web site

- Check restrictions about crawling – *robots.txt*
- Examine sitemap file - links to all Web pages
- Estimate Web site size – determines efficiency of crawling (Web site with million of pages requires distributed downloading)
- Technology used – static/dynamic content and interactive determine how we crawl

HTML tags in Web pages

- **<!DOCTYPE html>**: type declaration
- Contained between **<html>** and **</html>**
- **<head></head>** contains header information
- **<body></body>** contains visible part of page
- Paragraphs start with **<p>** and links with **<a>**
- Tables are defined between **<table></table>** where rows start with **<tr>** and columns with **<td>**

```
<!DOCTYPE html>
<html>
<head><title>Wiki</title>
</head>
<body>
<p><a href='link.html'>Next</a></p>
<table>
<tr><td>a</td><td>b<td></tr>
<tr><td>c</td><td>d<td></tr>
<table>
</body>
</html>
```

Web scraping in Python

- Python provides several libraries for scraping Web data
  - mechanize, Scrapemark, Scrapy, regular expressions, lxml, *BeautifulSoup*
- BeautifulSoup is a popular module being used for Web scraping, since it is easy and intuitive
  - *Urllib2* standard Python module can be used in combination with *BeautifulSoup* for fetching Web pages

Web scraping using BeautifulSoup

- Import libraries
  - *import urllib2*
  - *from bs4 import BeautifulSoup (https://www.crummy.com/software/BeautifulSoup/)*
- Specify the URL of page to be scraped
  - *page = urllib2.urlopen('http://www.akc.org/content/news/articles/labrador-retriever-is-once-again-americas-most-popular-dog/')*
  - *soup = BeautifulSoup(page)*
- Structure of the page – *print soup.prettify()*

Web scraping using BeautifulSoup (2)

- soup.<tag> returns content between <tag></tag>
  - *soup.title - <title>Breaking News:Labrador Retriever Is Once Again America's Most Popular Dog – American Kennel Club</title>*
  - *soup.a - Home*
- soup.<tag>.string returns string within tags
  - *soup.title.string – u`Breaking News:Labrador Retriever Is Once Again America's Most Popular Dog – American Kennel Club'*
  - *soup.a.string – u`Home'*
- Find all links in a page
  - *soup.a – provides only one link*
  - *soup.find_all("a") – provides all links in the page*

Web scraping using BeautifulSoup (3)

- Find tables
  - *soup.find_all('table') – extract information from all tables*
  - *my_table = soup.find_all('table')[0]*
- Extract information to Pandas DataFrame

```
for row in my_table.find_all('tr'):
    col = row.find_all('td')
    A.append(col[0].find(text=True))
    B.append(col[3].find(text=True))
    import pandas as pd
    df = pd.DataFrame(A, columns=['Breed'])
    df['2013'] = B
    print df
```

Breed	2013
Retrievers (Labrador)	1
German Shepherd Dogs	2
...	...
...	...

Geocoding of data

- Extracting geographical data (longitude and latitude) from address data (text)
- Required in spatial data analysis and mining

- Examples: Health epidemiological research such as finding local clusters of a disease, or analysis of geographical health issues, tracking, and marketing

Ways of geocoding

- Using geocoded reference dataset, addresses are matched to corresponding coordinates
- APIs
  - Several APIs are available to enable geocoding (Example: Google maps, Yahoo maps, Geocoder, etc.)
- Python
  - Python geocoding libraries (GeoPy, geocoder)

Address text

- Multi-variate attribute
  - Unit number, street number, street name, suburb name, state, postcode, country
  - Example: Unit 20, 18 North road, Acton, ACT 2602
- Exact matching
  - Exact location by one-to-one match
- Fuzzy matching
  - Possible matches (one-to-many) in neighboring region

Forward and reverse geocoding

- Geocoding is a two-way process: forward and reverse mapping
- Forward geocoding
  - Transforming address (text) data to geographical coordinates (longitude and latitude)
- Reverse geocoding
  - Transforming geographical coordinates to address data

Geocoding with Python

- Using Google's geocoded data

  import geocoder (https://pypi.python.org/pypi/geocoder)

- Forward geocoding

  g = geocoder.google('108 North road, Acton, ACT 2602')

  print g.latlng, g.city, g.state, g.street, g.country

- Reverse geocoding

  g = geocoder.google([-35.27,149.12],method='reverse')

  print g.city, g.country

Summary

- Extracting and transforming complex and semi-structured data, such as Web data and geographical data, is required in many applications
- Web scraping transforms Web data (unstructured) to structured data
- Geocoding transforms address data (text format) to geographical coordinates or vice versa