CPSC 490 Senior Project:

Practical and Yale-Affiliated Applications of Web Scraping Using BeautifulSoup

Joanna Wu (jw2375)

Advisor: Professor James Glenn

Spring 2020

Sections:

Acknowledgement

Many thanks to my family and friends for all of their support, academically and otherwise. Special thanks to my advisor, Professor James Glenn, for allowing me to take his classes semester after semester, the Yale Computer Science Undergraduate Learning Assistants, in particular Katherine Tan, who have provided so much encouragement, help and support in my CS courses these past 4 years when I doubted myself, and my good friends Aadit Vyas and Amrit Rau, who have helped me stay afloat when I was drowning. Thanks to Yale for the opportunity to learn and grow, I am forever grateful.

Abstract

In fulfilling my personal and academic goal of learning a new skill through my senior project, I conducted a deep dive and investigation into the power and ability of Python scripting, specifically in its web scraping and web crawling applications.

I began by building a web scraping program that would serve the needs of my a cappella group, which tours at domestic and international private and public schools, charging a performance fee to fund the costs of travel. A big part of the responsibilities of the group is to research schools in the area, and create a “call list” with contact information for each of the schools in the area. I set out to create a python script that could aggregate that data and create a call list, to learn the basics of web crawling. After creating the script, I reached out the Yale Undergraduate Admissions Office to see if they had similar “mundane” tasks that required time on behalf of the Admissions Officers, that I might be able to similar write a script for, and produced a series of scripts that replaced or supplemented their required work. Finally, throughout this project, I created a series of short write-ups and research reports on different tools, frameworks and concepts which could be used to assist in the comprehension of web pages, including Python’s Beautiful Soup library, web scraping methods and techniques, web

crawling prevention methods and techniques, semantic annotation of web pages, and computer vision based methods for intelligent parsing of web pages.

Deliverables

- Research write up on Python Beautiful Soup library and web scraping techniques
- Initial proof of concept script for Washington DC public schools
- Report and recommendations for Yale affiliated offices requests for scripts, their needs
- Script(s) written for the Yale Undergraduate Admissions Office
- Write-up on Computer Vision web page analysis and semantic annotation research
- Final project write up

Research Summary

- **Python BeautifulSoup and Common Scripting Techniques**
    - BeautifulSoup seems to be the primary Python library used by most for web scraping
    - Data can be exported to many different formats, including an excel spreadsheet/csv
    - Requests is used to send and receive URLs
    - First web scrapers were believed to be created in 1993, to serve in a search engine
    - There are multiple ways to accomplish the goal of scraping uniform data from a web page:
        - Copy and paste
            - Slowest and most labor intensive
        - Text pattern matching
            - Can use regular expressions

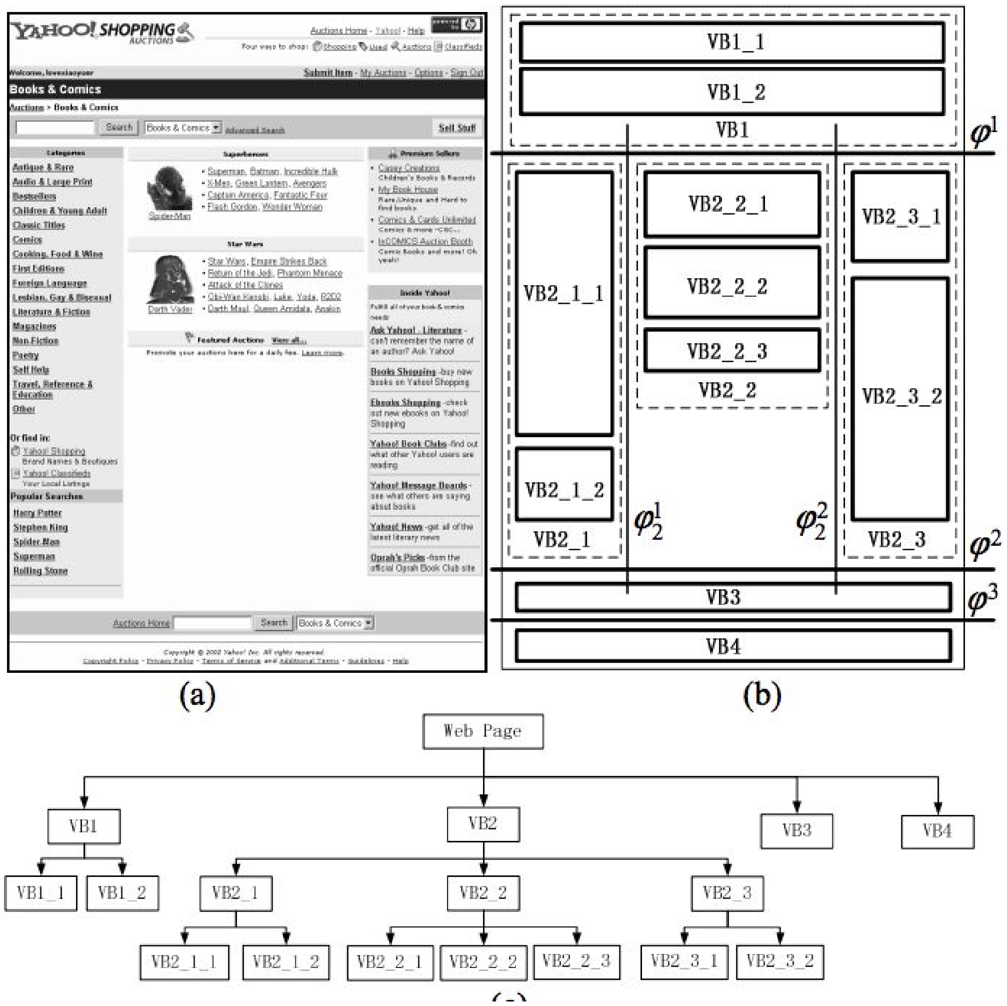
        - HTTP programming
            - Works with static and dynamic web pages, using socket programming
        - HTML parsing
        - DOM parsing
        - Vertical aggregation
            - Domain specific harvesting, very scalable
        - Semantic annotation recognizing
        - Computer vision web-page analysis
            - Attempts to use machine learning and computer vision to interpret pages visually
    - Cloud versus local
        - Local web scrapers can run on your computer, not very scalable
        - Cloud based allows for more simultaneous tasks
    - Some common uses of web scraping
        - Price monitoring
        - Market research
        - Mundane / repetitive tasks
        - Gathering contact information
        - Real estate
        - Sentiment analysis
        - News and content monitoring
    - Preventing web scraping- there are also multiple ways in which web scraping can be prevented by websites
        - Legal statement
        - Use a server that has a firewall built in, to protect your server from attacks
        - Use CSRF tokens

            - A hidden form field to make it harder to parse data on the website
        - Prevent hotlinking
            - This makes it so that if a link or image is taken off of the website, it won’t be displayed
        - Blacklist specific IP addresses
            - Useful if you can identify IP addresses which have been used for scraping
        - Throttling requests
            - Limiting the number of requests from one IP address
        - Change website structure frequently

- **Yale Offices Recommendations**
    - I contacted the Yale Undergraduate Admissions office and spoke to a Head Admissions Officer about some of their work that might be able to be replaced or supplemented by a script.
    - She mentioned that every year the admissions has to update their directory of all of the department DUS names and emails.
        - I looked into this and was able to scrape the information successfully, but it did require a bit of finagle-ing, because I didn’t know how to separate the departments and the directors by department, so I had to do that part manually.
        - I expected there to be more prevention of web crawling/scraping on the official Yale website, but there wasn’t.
    - They have to keep an updated list of student organization websites and officers.
        - I first tried this with the official college dean’s office list of student organizations, and there were some hidden displays that prevented me from being able to scrape the site.

        - However, I looked around some more and found that the admissions office already had a section with a limited subset of student groups, so I crawled that and used that.
        - Similarly to the issue with the departments, I couldn't find a smart way to categorize the student groups/links into the categories of the type of group they were, so I left that out of the spreadsheet.
    - She mentioned that sometimes they have to find/update every mention or usage of a certain thing on their website, which can be hard because they have lots of pages. An example of this would be when they changed "freshmen" to "first year".
        - I decided this wouldn't be the best use of my time to look further into / try to write a script for, because testing would require the ability to make changes to the site.
    - She mentioned that sometimes she would look at lists of award recipients for STEM competitions to see who in their applicant pool or accepted students pool had been awarded or recognized, and that it would be easier to have a spreadsheet.
        - She directed me to a few different websites to look into:
            - https://www.societyforscience.org/regeneron-sts/2020-finalists/
            - https://www.societyforscience.org/isef/awards/
            - https://www.cee.org/news/rsi-welcomes-82-top-scholars-2019-summer-stem-institute
- **Computer Vision and Semantic Annotation**
    - Computer Vision web page analysis examples
        - "Extracting Content Structure for Web Pages based on Visual Representation"
            - Web pages these days are really complex, contain many elements and different topics

- Researchers have been using many different techniques, from database techniques to semantic parsing to topic distillation to segment the pages
- Proposes a VIPS (Vision-based Page Segmentation) algorithm to determine structure
- Basic object is the leaf node in the DOM (document object model, they also have layout blocks which are groups of basic objects
- Each level in the DOM tree is parsed one by one, to check if the current object is a single block. If not, then it’s children are parsed. When parsed, the blocks are put into a pool and checked for visual separators. Putting the content structure back together is the creation of the content structure



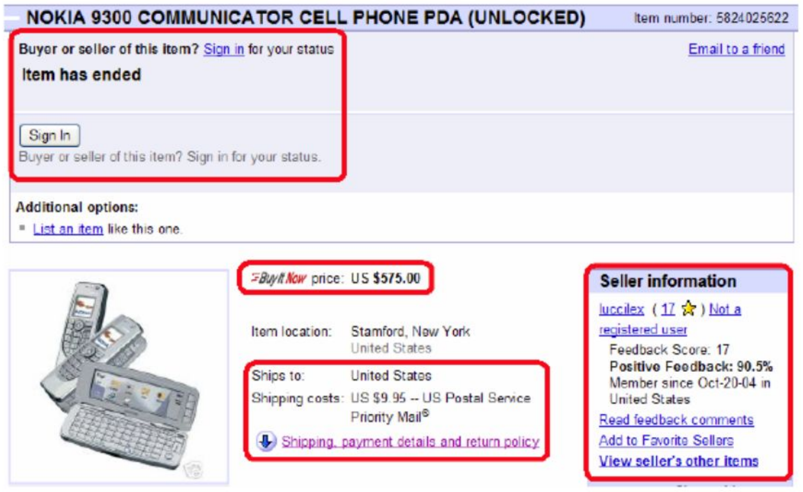
- an example using Yahoo
- Algorithm is efficient, uses top-down analysis
- When tested against human analysis, 97% of pages were corrected detected
- Next steps: adaptive content to consider mobile usage

■ "Computer Vision-based Analysis of Web Page Structure for Assistive Interfaces"

- Goal oriented around better web experiences for impaired users
- Uses two algorithms- segmentation and classification algorithm

            (a) Segmentation: recursively divide image into tree
            (b) Classification: label regions of trees into roles on page
            (c) “Divide and conquer” method
            (d) Differs from VIPS method because of this method, uses image of the rendered page as evidence, uses Bayesian framework over hand-coded heuristics
            (e) Advantages: can be more accurate to the intention of a web page designer than the look of the internal code which is created for functionality
            (f) ARIA friendly
    - “Experimental web-scraping using the Google Cloud Platform”
        - They created an experiment flow by using images of websites, labelling the data on the websites, training a model on the data, and then evaluating the predictions made on that data and extracting information
        - Results: 96.88 precision
        - Advantages: use of an API makes building features in the future faster, is more accessible, and a vision based approach is easier to maintain
- Semantic Annotation
    - Benefits: Content reuse and implication of new knowledge
    - Gather text, use NLP on the text to split sentence and tag parts of speech, classify any unique entities, make connections and recognize relationships between known entities, represent knowledge in a framework, stored in a database. Can be combined with other data sets of knowledge
    - Advantages: reduced operational costs, smarter search and more complex queries, better representation of knowledge, more durable and long lasting storage of knowledge

- Similarly requires split training and testing data
- “Semantic Annotation of Web Pages Using Web Patterns”
    - Adding metadata knowledge to web content
    - Goals of simplifying querying and improve relevance of answers
    - Relies heavily on the intuition of work of web designers, to fulfill and mimic user goal
    - Classify web patterns as organization of user controls, other common UI components
    - This paper focuses on web pages selling products



NOKIA 9300 COMMUNICATOR CELL PHONE PDA (UNLOCKED) Item number: 5824025622

Buyer or seller of this item? Sign in for your status

Item has ended

Buyer or seller of this item? Sign in for your status.

Additional options:

List an item like this one.

Buy It Now price: US $575.00

Item location: Stamford, New York United States

Ships to: United States

Shipping costs: US $9.95 -- US Postal Service Priority Mail®

Seller information

luccilex (17) Not a registered user

Feedback Score: 17

Positive Feedback: 90.5%

Member since Oct-20-04 in United States

Fig. 1. A web page with marked patterns. The patterns found are graphically marked on the page: *Sign on possibility*, *Price information*, *Purchase possibility*, and *Rating*.

    - Elements are evaluated by proximity, similarity, continuity and closure
    - Simplify querying by breaking queries down into key words
    - Created a data set, and ran algorithm: taking the plain text, and data entity, then comparing to create pattern.
    - Based on plain text, not HTML which is a big advantage
    - Key characteristics of web patterns are independent of language environment

Conclusion

Through researching and learning about Python's BeautifulSoup and scripting methods, I was able to go from knowing nothing about web scripting to being able to create a CSV with all of the public school names and principal's emails addresses for Washington D.C. Through working with the Yale Undergraduate Admissions Office, I was able to continue learning more scripting techniques, work with edge cases, create more useful scripts, and learn more about how websites prevent scraping. Finally, through researching semantic annotation and computer vision, I was able to better understand different methods of intelligently parsing a web page, and the goals, limitations, and challenges of each method.

References

Washington D.C. Schools Website: https://dcps.dc.gov/

Lists of Undergraduate Organizations at Yale:
https://admissions.yale.edu/extracurriculars,
https://studentorgs.yalecollege.yale.edu/directory

Yale Undergraduate Admissions Office Website: https://admissions.yale.edu/

Regeneron Awards Websites:
https://www.societyforscience.org/regeneron-sts/2020-finalists/,
https://www.societyforscience.org/regeneron-sts/2020-scholars/

ISEF Awards Websites:
https://www.societyforscience.org/press-release/16-year-old-engineer-works-to-improve-spinal-surgery-using-machine-learning-and-computer-vision-full-awards/,
https://www.societyforscience.org/press-release/intel-isef-2019-special-awards-winners-announced/

Kudelka, Milos & Snasel, Vaclav & Lehecka, Ondrej & El-Qawasmeh, Eyas & Pokorný, Jaroslav. (1970). Semantic Annotation of Web Pages Using Web Patterns. 10.1007/978-3-642-01350-8_26.

Cai D., Yu S., Wen JR., Ma WY. (2003) Extracting Content Structure for Web Pages Based on Visual Representation. In: Zhou X., Orlowska M.E., Zhang Y. (eds) Web Technologies and Applications. APWeb 2003. Lecture Notes in Computer Science, vol 2642. Springer, Berlin, Heidelberg

Michael Cormier. 2016. Computer vision-based analysis of web page structure for assistive interfaces. In Proceedings of the 13th Web for All Conference (W4A ’16). Association for Computing Machinery, New York, NY, USA, Article 24, 1–2. DOI:https://doi.org/10.1145/2899475.2899506