Fundação Getulio Vargas

Mestrado em Modelagem Matemática

Prof. Jorge Poco Due: July 30 2019

Homework 1

Instructions:

- Create a private GitHub repo named FDS-Homeworks in your local account. Add me and the TAs as collaborators (jpocom, jaoxvalen, eliorodriguez).
- Send me the link to the repository by Friday July 19 to my email.
- Create the folder hw1 in your repo and add all the required material for this homework there.
- Create a TAG named FINAL-HW1 and post the link at Google Classroom by July 30.

In this assignment, you will build a web application that scrapes https://scholar.google.com/ for data to authors and papers. Then display the information in a single HTML page. The following outlines what you need to do.

Step 1 - Scraping [4pts]

Complete your initial scraping using Jupyter Notebook, BeautifulSoup, Pandas, Requests, Selenium and Altair.

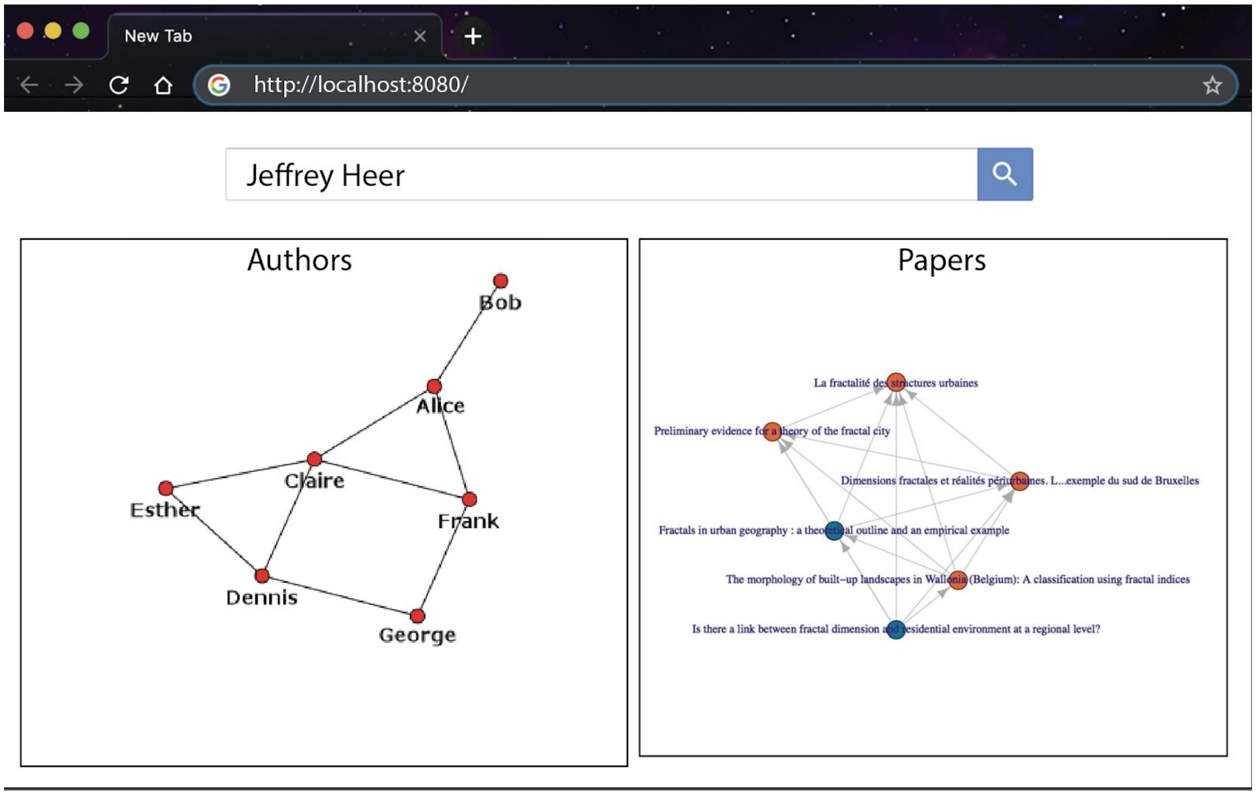
- Create a Jupyter Notebook file called scholar.ipynb and use this to complete all of your scraping and analysis tasks.
- Visit the url for Google Scholar here.
- Use Splinter to navigate the site and make a query with an author name, for example write Jeffrey Heer. If the author exists, then it should appear in the top part side, a link to the author's page. In this case, it will take us to the following link.
- Now you must scrape all the author's papers and coauthors. Note that for some papers the list of authors can be too long and Google Scholar shows for example "J Poco, R Etemadpour, FV Paulovich, TV Long, P Rosenthal, ...". You can omit the "..." and the remaining authors, in this case, keep only the 5 authors shown.
- You need to scrape all author's paper, so be sure to click "SHOW MORE" until this option is disable.
- You output should be an array like this:

```
# Example:
papers = [
    {"title": "D³ data-driven documents", "authors": [“M Bostock”, “J Heer”]},
    {...},
    {...},
]
```

Step 2 - Database & Visualization [6pts]

Use SQLite with Flask templating to create a new HTML page that displays all of the information that was scraped.

- [1pt] Start by converting your Jupyter notebook into a Python script called scrape_scholar.py with a function called scrape(author_name) that will execute all of your scraping code from above and return one Python array containing all of the scraped data.
- [1pt] Next, create a route called /scrape?author=Jeffrey+Heer that will import your scrape_scholar.py script and call your scrape(Jeffrey Heer) function.
  - Create three tables in SQLite: author, paper and author_paper. Authors and papers should have unique values. author_paper table will help us to link papers to authors, this table should have the ID from authors and the ID from papers. If you need more details read for many-to-many relationships in relational databases.
  - Store the return array in SQLite. Every query should increase our tables by appending new records.
- Create a root route / that will be your main page.
  - [1pt] Create a template HTML file called index.html that will help us to query a new authors and display our data. Use the following as a guide for what the final product should look like, but feel free to create your own design.
  - [1pt] In the top bar, we can enter a new name and our app should scrape the authors name (if it doesn’t exist yet) and populate our database. Immediately it should update the graphs below.
  - [2pt] There should be two panels, one for authors and one for papers. Edges between authors should exist if the authors share at least a paper. Edges between papers exist if they have common co-authors.
    - For the node locations in the screen you can use the force layout algorithm from here.
    - Use the https://altair-viz.github.io/ functionalities to select authors and highlight the his/her papers and vice versa.
    - Show the authors’ names and papers’ title by hovering the nodes.


New Tab
http://localhost:8080/
Jeffrey Heer
Authors
Bob
Alice
Claire
Esther
Frank
Dennis
George
Papers
La fractalité des structures urbaines
Preliminary evidence for a theory of the fractal city
Dimensions fractales et réalités périurbaines. L...exemple du sud de Bruxelles
Fractals in urban geography : a theoretical outline and an empirical example
The morphology of built–up landscapes in Wallonia (Belgium): A classification using fractal indices
Is there a link between fractal dimension and residential environment at a regional level?

Hints

- Use Splinter to navigate the sites when needed and BeautifulSoup to help find and parse out the necessary data.
- Use sqlite3 for CRUD applications for your database.
- Use Bootstrap to structure your HTML template.