STA 141B Project Report

Joseph Gonzalez

11/30/2020

Project 3: Webscraping

The Websites And Terms Of Service

Goal: To scrape Indeed and Cybercoders.

Terms of Service

Before web scraping, we first look at the terms of service for each website. Many websites do not want their data scraped and replicated. To avoid legal troubles and website banning, we want to avoid violating the terms of service. After reading through both Indeed's and CyberCoders' terms of service, it seems that indeed is more strict on web scraping than Cybercoders. The statement that stands out for both Indeed and Cybercoders is that they do not want their data to be replicated, sold, and traded. Therefore, it is important to mention that this project is only for academic purposes and will not be used to replicate or publish scraped data to another website.

Website Layout

Both websites are job boards that list jobs for different positions within various companies and industries. We first explore both websites and see the possible patterns we can use to get the data we need. Another detail to be aware of is how the URL changes for new searches or a new page(farther down the job list).

For the most part, the website's job pages seem to have a structure for job tasks, location, salary, and preferred skills. We can use Xpath commands to get this information. In other cases, the information may be missing or not entered in the designated regions. We can use conditional statements and grep expressions to correct these situations.

Scraping The Websites

Scraping CyberCoders

First, we query the results for the search term, like *Statistician*, *Data Analyst*, and *Data Scientist*. We next go into the developer tools and copy the URL for the search page. We insert the URL into the function getForm() to obtain the source file for the search webpage and parse the object using htmlParse(). Next, we use XPath operations on the parsed object to obtain the job names and the URL to each job page. After we obtain the job name and specific URL, we immediately scrape(using getForm()) the job source page and store them as a character element in a vector. We do this for every job on the search HTML page. After

collecting the details for the jobs, we use xpath to search for "next" to get the URL for the next page of listed jobs. We repeat this process until there are no further pages. This procedure provides us with an n by 2 matrix(n is the total number of job listings) with job name as column 1 and the string HTML page as column 2. We can use the second column to obtain more specific data, like salary or location, about the jobs and employers.

Scraping Indeed

Compared to CyberCoders, there seems to be(at least in my case) more issues related to scraping the Indeed data. I first used the getURL and getForm functions to try and get the searches' HTML page. However, the output object came back empty. I used the parameters verbose and followlocation to resolve this issue, but this resulted in an object that contained only numbers. I later found that readLines obtained an HTML page and, unfortunately, this page had search jobs that were out of order(Some jobs on the page did not appear on the first search page). I eventually found that entering the request cookie, referer, and user-agent into the .opts parameter for getForm() provided the correct html page. Since Indeed has a strict policy on scraping, I believe these fields are required to get the source page content.

After this issue, I took the same approach to web scrape Indeed as Cybercoders. I used XPath expressions to scrape the job names and URLs on the initial search page. I immediately retrieved each job's source page and then acquire the URL for the next page in the search query. I repeated this process and stored the information in an n by 2 data frame.

Update: I used a city(Oxnard, CA) close to my home to narrow down the searches. After a certain amount of job scraping, I run into the reCaptcha issue. I used Sys.sleep to alleviate the issue, but this can extend the amount of time for my code to scrape all the posts. On a small number of job posts, my code worked and I assume it can work for a large search query.

Scaping precaution

Some searches query one job page or no results. To resolve this, we can set conditional statements that will check for missing next page URLs and missing entries. I also encountered a recaptcha issue on Indeed. This issues arises from the website detecting an automated system scraping the page., To resolve this, the request cookie must be an argument in getForm's .opts parameter.

Obtaining Job Related Data

CyberCoders

For each job, the job information is located in designated areas on the CyberCoders' web page. To obtain the data, we first search on the source page for the node these items are on. For example, the job location is in the node set that has class = "location" and the salary is in a node set that has class = "money." We use these patterns to extract the web page's text(using xpathSApply) and we apply this procedure to every job web page. An important note is that not all information are in these fields and, therefore, it is important to have conditional statements to check for missing entries. Another important remark is that not all the information we want is in certain designated areas. Therefore, we utilize regular expressions and regular expression functions(grep, grepl, gsub, etc,) to obtain this information. Another way to get information would be to split the full text into a character vector and search for the specific terms we are looking for(use strsplit).

Indeed

Web scraping for job-specific information on indeed would be relatively similar to CyberCoders. However, the format for indeed posts are messy and varies from entry to entry. This means that we would rely

much more on regular expressions on indeed than on CyberCoders. We would also need more conditional statements for any "what-if" situations.

Improvements

The code to obtain job-specific information can be further improved. While the code does check whether the job information is missing in designated spots, it doesn't search through the entire text for information that in irregular spots. Therefore, I would implement more conditional statements and regular expression functions to find the information that are in unusual locations.

Prompt Questions

This results description/analysis is related to the CyberCoders' data.

On the CyberCoders website, I searched for data scientist(10,876 on Indeed), data analyst, statistician, and data engineer. There are 16 posts available for data scientist(10,876 total on Indeed, 19 close to Oxnard, CA), 28 posts available for data analyst(16,385 on Indeed, 35 close to Oxnard, CA), 138 posts available for data engineer(115,364 on Indeed), and 0 posts available for statistician(1,877 on Indeed, 5 close to Oxnard, CA).

For each job search, I created an n by 13 data frame. Each row is a job post and the columns contain information about the post. More specifically, the columns contain the job name, html document(string), all included text in the post(No headers), job tasks, job location, salary range, employment type, required skills, preferred skills, education, education field, years of experience, employee benefits, and company incentives. If a field was missing or not indicated properly, the entry will indicate that there is no specified information.

The data varies significantly. Some posts may have more than sufficient information and others may have no information at all(at least job title and location). There are also times where some posts have different job description sections than other job posts(Some posts have benefits sections). For the queries, we can see commonalities with the employment type, education level, and degree fields. For example, the jobs often require at least a bachelor's degree and a technical degree(computer science, statistics, engineering). The main differences between the title queries are the locations, requirements, and preferred skills. The locations vary across the United States. The requirements and preferred skills seem to depend on the job and company. Some jobs require knowledge of many computer softwares and others require more personal attributes(organization, attention to detail, etc.).