Working on the Scraper

The scraper is our primary method of getting data. It's important we keep this up to date because we'll practically always want to be scraping more training data.

What the system looks like now

Currently, we have one scraper that is multi-threaded and will run on about 10 different job titles to collect information from just one site: Glassdoor.

We usually leave the scraper running overnight. Each process creates its own csv file, which you can find under "output_csvs". From there, we run the file "/scraper/coallescer.py" to collect them all into one output csv.

We then have to clean and tokenize the data. This means we're trying to strip the non-words out (spaces, punctuation, etc). Run the functions under the data_processing folder and that will produce the final csv from which we all grab data that we want to tag.

In Depth: Glassdoor_scraper.py

The meat of this file is in the get_jobs function. Here, we open up a chrome tab, try to close the pop-up, and then begin our scraping.

For each listing on the page, we attempt to find information like the company name, job title, and description. All of these are optional except the description itself, so for now we've ignored the others. For the job listings we have, we append them to our current working list and then use an ActionChain to click on the next listing. This is important because the descriptions don't load until you actually click the listing in the sidebar.

Notes:

- You'll need to keep up-to-date on the xpaths to the job listings. They often change, and when something breaks, you should check these things before you try to change anything else.
- Note that our action chains are currently set up to find the pop-up by position. That's because often, these elements are not listed on the page easily, and Selenium will only allow you to click top-level items. One of the ways Glassdoor foils us is by making sure the "x" is not top-level. This is how we get around them. But this technique may work for other processes also - keep it in mind 😃

Current Issues

- Biggest obstacle is that our scraper stops after a few dozen pages. We think this because pop-ups keep coming up at unexpected times, although some investigative work is necessary here.
- Another issue is that there are sometimes issues moving onto the next page is you've reached the end of the slider. E.g.: you're at page 10 and the directory at the bottom lists pages 1-10. It's hard to detect what's going on in this case, but has to do with the fact that the elements in the page change every time. We've managed to fix the issue so we do get past a fair number of pages, but this needs to be checked out and standardized.
- Lastly, we need a good process for organizing the data in such a way that we can each take a portion every week and do our tagging without ruining the structure for everyone else.

Eventual Goals

- Opening up the scraper to other sites would mean we might get a wider variety and greater amount of job listings. Not a priority since massive amounts of cross-posting mean you get much of the same data.
- All-in-one pipeline for data processing: one batch script that sets off the scraper, does the data processing, and prepares the data for hand-tagging.

Steps you can take

Priority (/5)	Expected Difficulty (/5)	Description
5	5	**Extend scraper**: modify scraper so that it can keep going endlessly. Requires investigative work to determine which pop-ups or other factors are stopping it and re-writing the scraper module to account for that. Huge impact, this is the priority once you feel comfortable with the scraper.
5	1	**Creating process**: design a process for multiple people to be able to pick, tag, and organize resulting data into once place. This requires looking at how the data is used in the spacey model, but is mostly about organization and communicating the new process in such a way that everyone can follow the same process easily.
4	2	**Batch script**: create a script that automates everything from collection to processing of the data. A little difficult because you have to account for running the scraper overnight, but shouldn't be too difficult to set up.
3	4	**Pagination errors**: modify scraper such that getting onto the next page is no longer an issue. Requires looking at the patterns of where page elements change after reaching the end of a section of pages and devising a method for getting past it.