Data collection – API Exploration and Web Scraping Strategies

AP Research & Senior Research Project

Ojas Chaturvedi

https://github.com/ojas-chaturvedi

This week, I've progressed with my data collection efforts for the sentiment analysis models. As outlined in my previous blog post, my objective was to leverage the ProPublica Congress API to identify legislation pertaining to the term 'gun'. Initially, I anticipated that the API would grant access to both metadata and the legislation text. However, upon testing this week, I discovered that accessing the text required a premium subscription, an option I reserved for later consideration if necessary.

During my exploration of the legislation metadata, I noticed a property field containing a link redirecting to the legislation details on Congress.gov. Upon further investigation, I learned that Congress.gov also provides an API per request, which I successfully obtained access to a few days ago. Moreover, I discovered that each legislation page includes a designated tab offering the text, accessible via the URL structure. For example, appending "/text?format=txt" to the legislation link, such as in the case of H.R. 7544:

Original link:
https://www.congress.gov/bill/117th-congress/house-bill/7544

Modified link:
https://www.congress.gov/bill/117th-congress/house-bill/7544/text?format=txt

This modification grants access to the full text of the bill. With this realization, I could utilize the Congress.gov API to identify relevant legislation and employ a URL script to access the complete text.

However, the challenge lies in extracting the text from the website. This process required considerable effort and ingenuity. After analyzing multiple sites and scrutinizing their page source code, I discovered that the text consistently resides within a <pre></pre> tag in the HTML structure. Consequently, I have started working on creating a web scraper to access this tag and retrieve the legislation text. To achieve this, I've been understanding the documentation of Python libraries such as Selenium and Beautiful Soup, and successfully implemented the scraper. Here is the GitHub link to access the code for the scraper: **https://github.com/ojas-chaturvedi/NLP-Gun-Legislation/blob/master/web-scraper.py**

During my internship at the ASU Data Mining and Reinforcement Learning Lab, I have finished understanding the foundational paper, “Safe Reinforcement Learning with Natural Language Constraints”, serving as the bedrock for my current internship project [1]. Safe reinforcement learning entails integrating safety constraints into the reinforcement learning process, enabling intelligent agents to achieve the most optimal results while following the constraints. This paper uses natural language constraints — textual constraints replacing conventional mathematical or logical equations. Unlike instructions that specify actions, textual constraints outline actions to avoid, independent of maximizing rewards. Since constraints are decoupled from rewards and policies, agents trained to understand certain constraints can transfer their understanding to respect these constraints in new tasks, even when the new optimal action is drastically different. Despite its transformative potential in the reinforcement learning field, this concept remains largely unexplored, with the paper being rejected due to its limitations in contributions and reproducibility. Therefore, my internship project will capitalize on this idea and craft a new experiment and subsequent paper. Currently, my focus lies in identifying efficient benchmarking environments conducive to simulating both agents and obstacles. These environments are crucial for rigorously testing the agents’ capacity to adhere to specified constraints and subsequently transferring this proficiency to new environments.

In the week ahead, my main focus remains finalizing data collection. Recently, I gained access to the ASU supercomputer, Sol, crucial for executing my web scraper that needs to iterate through over 1500 legislation links. Given the impracticality of executing this task on my personal computer, I plan to integrate the web scraper with the API, encompassing all legislation links, and familiarize myself with Sol’s infrastructure for efficient deployment. Before I send in the script to Sol, I will check the code with Saianshul, who also has coding experience, to make sure it is as time and space-efficient as possible. Additionally, based on a suggestion from Dr. Travis May, I’ll explore broadening the scope beyond ‘gun’ legislation by incorporating keywords like ‘firearm’ to capture a comprehensive view of the gun control discourse. Lastly, I am waiting for access to OpenAI’s Researcher Access Program, allowing me to categorize legislation texts into pro-gun rights or pro-gun control stances, a crucial step before conducting sentiment analysis.

[1] T.-Y. Yang, M. Hu, Y. Chow, P. Ramadge, and K. R. Narasimhan, “Safe Reinforcement Learning with Natural Language Constraints,” *Neural Information Processing Systems*, vol. 34, May 2021.