**CPI Affiliate Publications Tracker**

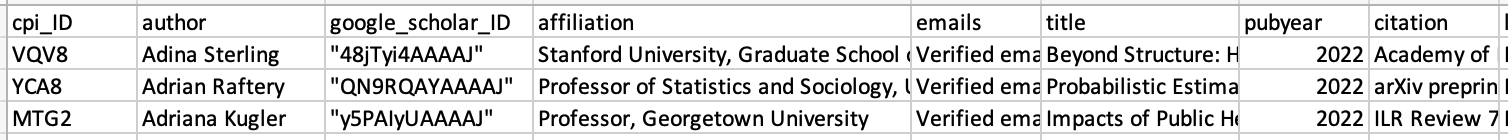
**Overview**

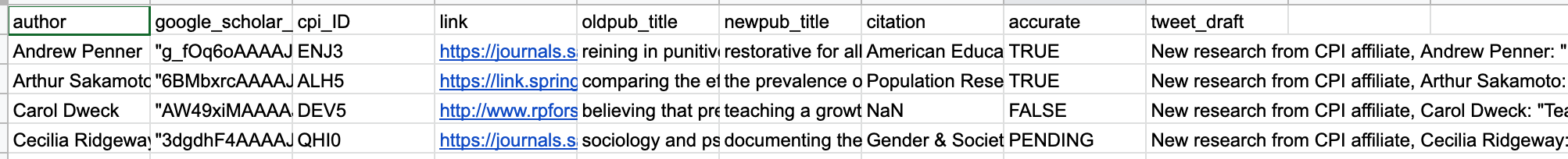
Our goal is to keep track of new publications by CPI affiliates which will be announced through daily tweets and a weekly newsletter. Because there are over 500 affiliates, we created an automated system using Google Scholar to help us do this.

Code can be found at:

**Summary of steps**

We have a data file with relevant information for each affiliate and their latest publication:



****Every day we will run a script to obtain the latest publication for each affiliate on Google Scholar. If the publication we find does **not** match the one in our dataset, we presume it’s a new publication and add it [this Google Sheet](https://docs.google.com/spreadsheets/d/1CiV3WkPRUkFcCfxmZI3xcQYx3ng3sjy8AfodqW_UbuU/edit?usp=sharing).

The Google Sheet has a column “accurate” which must be updated by a human reader who will check if the publication we found is correct. We will fetch this Google Form the next morning. For all publications verified as TRUE, we will:

1. Update our affiliate database to incorporate this new publication.
2. Use the Twitter API client to post a tweet about it. I did this from a personal account below to test it, but we can connect it to the real CPI twitter. Whoever verifies the publication on the Google sheet will also be able to edit the tweet draft if desired (standard format for the tweet can be seen below).

Graphical user interface, text, application

Description automatically generated

Publications verified as FALSE will be added to an “error” document to ensure we don’t add it to our database again in the future. All verified publications (accurate = TRUE or FALSE) will be removed from the Google Sheet so person does not have to do the same work twice.

**Detailed process**

Tools

Because Google Scholar does not have an API service, we rely on two different services to run daily searches for new publications:

* [Scholarly:](https://pypi.org/project/scholarly/) Python module to retrieve author and publication information from GS. Because Scholarly is free, we try to maximize the use of this service and minimize SerpAPI searches.
* [SerpAPI](https://serpapi.com/): Paid service that provides more advanced/flexible scraping from GS. Our current account plan gives us 5,000 individual searches per month.

Pre-processing

The first step in the process was to obtain the Google Scholar ID of every CPI affiliate. This ID is useful because IDs are unique, names are not. Using the correct ID is much more likely to yield accurate search results. We used SerpAPI to do an organic search of each affiliate’s name and retrieve the corresponding ID. For those affiliates we couldn’t find an ID for, we do a second check using Scholarly. We were able to find a Scholar ID for over 70% of CPI affiliates.

The second step is to create a database with the latest publication by each CPI affiliate. This data is essential because we will determine if a publication is “new” by comparing the latest title retrieved with the one in the database. We retrieve the latest publication for affiliates with a Google Scholar ID using Scholarly and fetch the rest using SerpAPI.

After creating this database, we carried out a manual check to verify the results made sense. If, for example, we obtained a publication from an Electrical Engineering journal for an education researcher, we inspected the result to ensure we had the correct google scholar ID.

Daily tasks

Every morning, we run a script to automatically:

1. Open and read the file *affiliate\_publications.csv* containing the list of publications for every affiliate
2. Open and read the file *tentative\_titles.csv* which contains the list of publications found on previous days which have **not** been reviewed by a person (status = pending).
3. Run a script to fetch the latest publication for each affiliate on Google Scholar.
   1. We will only fetch publications from affiliates not found in the tentative titles document. This is because we know these affiliates have new pubs and we don’t want to waste API calls fetching information from them again.
4. We compare the title obtained to the title in our file. If the titles don’t match, this becomes a “tentative” new publication which we append to the *tentative\_titles.csv* file.
5. Open and read the file *error\_publications.csv* containing publications a human reader has flagged as wrong (more on this below). We remove publications contained in this list from our tentative titlesdataset.
   1. We also remove articles from *error\_publications.csv* if they didn’t show up in the new publications list. This would indicate that the search for the author or author ID no longer returned the problematic result, and we can stop tracking it. We do this check so our error publications dataset does not become infinitely long with time.
6. For all remaining publications, we fill in any missing data:
   1. For each article, we get the following information: article title, publisher, and URL.
   2. Articles obtained using Scholarly don’t have an associated URL so we use SerpAPI to obtain it.
   3. Articles obtained using SerpAPI don’t have an associated journal/publisher. We have to use a different SerpAPI query to obtain it.
7. Finally, we save and export the tentative titlesfile.

At this point, our process requires human intervention. These are the steps:

1. Data from the tentative titles file is exported into a google sheet owned by the CPI bot google account I created (account information below). However, the sheet can also be shared with whoever will be editing it every day.
2. The Google Sheet contains all information about the publication including the author, title, URL, journal, and Tweet draft. It also contains a column titled “accurate” which will be set to PENDING.
3. Because there might be some false hits (e.g., pubs authored by a non-affiliate with the same name), a human will need to verify each row and update the “accurate” column to either TRUE or FALSE based on their assessment.
4. For accurate publications, a Tweet will be created with information from the publication. An example using data retrieved 08/07/2022:
   1. *"New research from CPI affiliate, David Neumark: "Help Really Wanted? The Impact of Age Stereotypes in Job Ads on Applications from Older Workers". Read at*[*https://www.nber.org/papers/w30287"*](https://www.nber.org/papers/w30287%22)
   2. The human reader will be able to directly edit the tweet draft in the Google Sheet if they want to make changes or replace the draft with the word “False” if they don’t want any tweets related to the publication. For inaccurate tweets, it is not necessary to edit the tweet as all drafts will be automatically discarded.
5. The next morning, we will fetch the verified data from Google Drive. For each article, we will check its status and take one of three steps:
   1. If the publication hit is accurate, we use the Twitter API client to create a tweet on the CPI account announcing each publication. We also update our main affiliate information document to reflect the latest found publication. QUESTION: should we tweet all publications out at once or space them out?
   2. If the publication hit is inaccurate, we append it to the database *error\_publications.csv.* We have this database because we know our search will likely return the same error file again (as we are doing the same search each time) and we want to avoid adding it to the list of potential titles and making the same mistake twice.
   3. If a publication is still pending, we keep it in the potential titles database. This way, the publication will be included in the next day’s Google Sheet for another chance to get verified.
      1. This step is just safeguarding in case a person does not edit the Google Sheet that day or does not finish going through all rows.

**Adding new affiliates**

Because the list of affiliates might grow over time, we also create a system to add new people to our list. The current way to add a new affiliate is to insert their name into a new row in [this Google Sheet.](https://docs.google.com/spreadsheets/d/1cNlU1zKcTTitx630vwFqqCMD6sv90NvniddBRNnRdTc/edit?usp=sharing) It is only necessary to fill out the ‘author’ field, though adding information for the other fields would be preferable (instructions below).

Every morning, we will also run a script to fetch data from this google sheet. For every new author, we will repeat the steps outlined in the pre-processing section to obtain their Google Scholar ID and affiliations. Then, we will incorporate them into the main affiliate spreadsheet that we use to query new publications.

Filling out additional fields

To get information for the other fields (google scholar ID, affiliation, publications, etct), we need to access their Google Scholar page. The simplest way to do this is by typing their name in quotation marks into the [Google Scholar](https://scholar.google.com/) search bar. If the person has a Google Scholar profile, a link to it should appear at the top of the page:

Graphical user interface, text, application, email

Description automatically generated

To get the latest publication information, click on their name to navigate to their scholar page. There, we can sort the publications by year and get information from the publication at the top of the list. We get their Google Scholar ID using the URL of their scholar page by finding the string of characters between “user=” and “&”. Using Sharkey’s url as an example:

https://scholar.google.com/citations?hl=en&user=J\_Df5rsAAAAJ&view\_op=list\_works&sortby=pubdate

**Account information**

To create this system, I had to make a few accounts. I have a separate document with relevant passwords which I have in a separate document (I didn’t want to write them down in a public instructions document).

* Google account: All relevant Google Sheets are saved in the Google Drive of a CPI bot account I made:
  + Email: [cpi.affiliate.scraper@gmail.com](mailto:cpi.affiliate.scraper@gmail.com)
* Google client: To access Google Drive from Python, we needed to create a Google client (using email above). OAuth keys for the Google client can be found in separate document.
* SerpAPI: We use SerpAPI to automatically query Google Scholar. To use the service, we need to include an API key in our code which can be found in separate document.
* Twitter account and client: I used a personal Twitter account to test out this code and had to create a developer Twitter account to obtain relevant client keys. We will have to do this again using the official CPI twitter account.

**Code and data files**

* *get\_google\_scholar\_id.py:*Finds relevant information for new affiliates including their scholar id and affiliations. It accesses list of new affiliates from ‘new affiliates’ google sheet.
* *find\_new\_publications.py:*Goes through every affiliate in our database, finds their latest publication available on Google Scholar, and compares it to our previously stored publication to determine if it’s new. The script then uploads potential new publications to ‘new publications’ google sheet.
* *process\_new\_publications.py:*This script grabs google sheet and processes publications based on the status assigned by human verifier. If a new publication is accurate, the script creates a new tweet announcing the publication and updates existing affiliate database to reflect newest publication. If it’s not accurate, the script adds the title to an on-going list of “error” publications to avoid in the future.
* [*new\_affiliates (google sheet)*](https://docs.google.com/spreadsheets/d/1cNlU1zKcTTitx630vwFqqCMD6sv90NvniddBRNnRdTc/edit?usp=sharing)*:* Every morning, we will pull from this google sheet to see if there are any new affiliates to add to our database. As such, any new members can be incorporated by writing their name into the file.
* [*new\_publications (google sheet):*](https://docs.google.com/spreadsheets/d/1CiV3WkPRUkFcCfxmZI3xcQYx3ng3sjy8AfodqW_UbuU/edit?usp=sharing)The sheet will contain the new publications we found. Because mistakes in the scraping will be common, a person will access this sheet and update the ‘accurate’ row to indicate if the find was correct or not.
* *cpi\_affiliates\_publications.csv:* Sheet containing all information from CPI affiliates including their latest publication.
* *error\_pubs.csv:* These are publications which previously yielded a false positive result. We check this data to make sure new publications we find haven’t already been flagged as inaccurate.
* *potential\_new\_pubs.csv:*List of potential new publications which we use to build out ‘new publications’ google sheet.

**Relevant resources used**

* <https://www.analyticsvidhya.com/blog/2020/07/read-and-update-google-spreadsheets-with-python/>
* <https://towardsdatascience.com/this-virtual-machine-is-saving-my-time-23076f592b94>
* [https://towardsdatascience.com/how-to-customize-your-pre-filled-google-form-using-python-c85ecc7dcb9](•%09https:/towardsdatascience.com/how-to-customize-your-pre-filled-google-form-using-python-c85ecc7dcb9)

To-do:

* Update list of affiliates (get names from Laura)
* System to save new pubs for newsletter
* Fix string bug (see tweet)