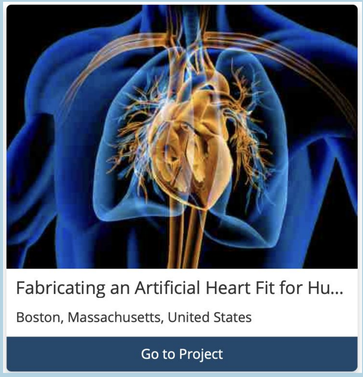
RePicture Data Engineering: Creating a Database to Help Students Access STEM Research

Christian Reyes Avina, Benjamin Liang, Mackenzie Moffit, Jordan Ta, Stephanie Xu
Data Science Discovery: RePicture
December 6, 2022

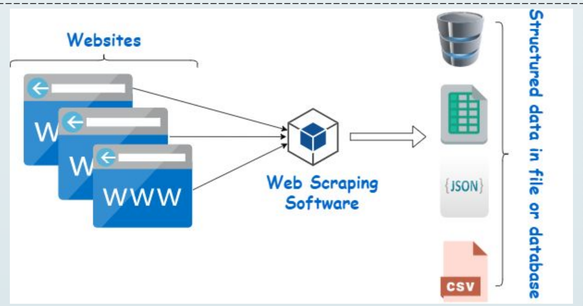
Problem

Our project was with the company RePicture. RePicture aims to provide resources to increase and spread knowledge of both STEM and careers in STEM. Our goal for this project was to create a database that conveniently presented STEM-related projects that were already on the internet. By doing this, students could have a better understanding of the STEM field they are interested in, what kind of research is done in that field, and who is doing the research. To do this, we would have to use web scraping to extract the information to put in the database.



This an example article from the RePicture Website STEM projects page. Our goal would be to offer articles like this on a large scale with our database.

Finding Data To Scrape



The first part of the process was finding the necessary data to implement a web scraping algorithm. Not all websites allow for webscrpaing, and due to legal precautions we had to make sure that data scraped was publicly accessible. The ScienceOpen repository was ideal for this because all articles published are publicly accessible and contained all the necessary data to build our database such as author, contributors, abstract, publication date, etc.

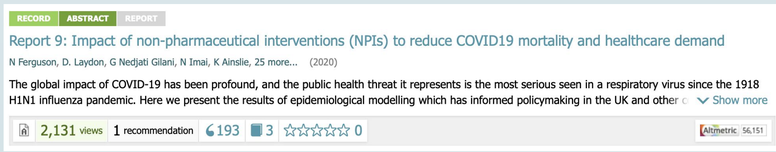
Scraping The Data

Once we chose the ScienceOpen website, we had to develop a program to scrape information from a large number of articles on the site. To do so, we used Python in Google Colab with the main libraries for scraping being BeautifulSoup and Selenium. We also used Pandas to put the data in a dataframe. We decided to gather from the article the following pieces of information: title, publication date, author(s), abstract, DOI, URL, journal, publisher, keywords, affiliations, and discipline. If an article did not have a one of more pieces of information, the field in the database would be left empty for that article.

We started by using BeautifulSoup to scrape the pieces of information by themselves. This library allowed us to select parts of the html of the article site and extract that information, which we could clean once it was extracted if necessary. After all the fields for the article are complied, they can be added to the pandas dataframe, where each row represents an article. We also used Selenium to press the "Load More" button on the ScienceOpen website so that we could scrape the URLs of all the articles we wanted to look it, and then further scrape the information from each of those articles. This was a main challenge of our project, so we spent a lot of time understanding Selenium. We also at one point realized that we had to click the "Accept Cookies" button on the ScienceOpen website before we could click the "Load More" button, so we also added code that would do this straight away.

We were able to scrape articles from multiple different disciplines. The ScienceOpen website has filters where you can select a discipline, such as Engineering, Computer Science, or Life Sciences. We also sorted by most to least cited articles, which would allow us to scrape articles that likely have useful and reliable content. We could take the URL after applying these filters and run it through our program to output the fields for articles with those filters in a dataframe. An example of a subset of this output for the discipline Medicine is shown below.

	Title	Author(s)	Publication Date	Abstract	DOI	URL	Journal	Publisher	Keywords	Affiliations	Discipline
0	Preferred Reporting Items for Systematic Revie...	[David Moher, Alessandro Liberati, Jennifer Te...	21 July 2009	Introduction Systematic reviews and meta-analy...	10.1371/journal.pmed.1000097	https://www.scienceopen.com/document?vid=4d10d...	PLoS Medicine	Public Library of Science		[Ottawa Methods Centre, Ottawa Hospital Resear...	Medicine
1	A new coronavirus associated with human respir...	[Fan Wu, Su Zhao, Bin Yu, Yan-Mei Chen, Wen Wa...	3 February 2020	Emerging infectious diseases, such as severe a...	10.1038/s41586-020-2008-3	https://www.scienceopen.com/document?vid=11e0d...	Nature	Nature Publishing Group UK	genetics, viral infection	[[Shanghai Public Health Clinical Center, , Fu...	Medicine
2	The Cochrane Collaboration's tool for assessin...	[Julian P T Higgins, Douglas G Altman, Peter C...	18 October 2011	Flaws in the design, conduct, analysis, and re...	10.1136/bmj.d5928	https://www.scienceopen.com/document?vid=2f4dc...	BMJ : British Medical Journal	BMJ Publishing Group Ltd.		[MRC Biostatistics Unit, Institute of Public H...	Medicine
3	Preferred reporting items for systematic revie...	[David Moher, Larissa Shamseer, Mike Clarke, D...	1 January 2015	Systematic reviews should build on a protocol ...	10.1186/2046-4053-4-1	https://www.scienceopen.com/document?vid=d35e9...	Systematic Reviews	BioMed Central		[Ottawa Hospital Research Institute and Univer...	Medicine
4	Global, regional, and national incidence, prev...	[]	10 November 2018	Background The Global Burden of Diseases, Inju...	10.1016/S0140-6736(18)32279-7	https://www.scienceopen.com/document?vid=39d4b...	Lancet (London, England)	Elsevier			Medicine



This is an example of an article on the ScienceOpen website.

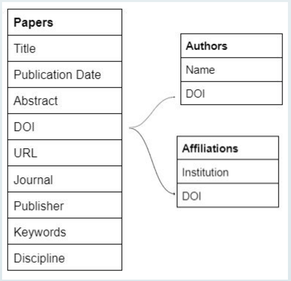
AWS

Because our script would be scraping hundreds of journals we had to use AWS in order to run the script for an extended period of time to later put into the database. We decided to use an EC2 instance in order to run our script and downloaded the appropriate packages to make it work. Specifically we needed to download a chromedriver and chrome in order to run the web scraping. The difficulty was ensuring that the chromedriver and chrome version were the same.

```
cd/tmp/
wget https://chromedriver.storage.googleapis.com/107.0.5304.62/chromedriver_linux64.zip
unzip chromedriver_linux64.zip
sudo mv chromedriver /usr/bin/chromedriver
chromedriver --version
```

MySQL Database

Pandas has a built in to_sql method which allowed for easy transfer of a dataframe to a sql database. We had decided to use MySQL database as we just needed simple storage and had not planned any manipulation immediately. We had two columns called authors and affiliations that included lists which cannot be stored on a relation database so we used the .explode() method in order to disperse the items into different rows. The data between authors and affiliations was not clear from the scraped data so future exploration is needed in order to find ways to fix this. Using the exploded dataframes one for authors and other for affiliations we exported the digital object identifier (DOI) in order to connect the authors and affiliations with the articles.



Conclusion

The use case for the data would be integration into the RePicture website for students to easily access journals they are interested in using the data scraped. For future improvements connecting authors and affiliations would be desirable. In addition scraping from more sources to get more journals would also improve the project.

Acknowledgements

We would like to thank Data Science Discovery for hosting the symposium and connecting us with companies to work with. We also would like to acknowledge RePicture for allowing us to work for them and Science Open for their collection of open access journals.