Instytut Literacki Web Scrapper **https://github.com/zdudzik/Kultura_Scraper**

Background & Overview

The Paris-based, Polish publishing house *Instytut Literacki*, founded with the help of the remains of the Polish 2nd Corps at the end of World War II, is considered one of the most important publishing houses of the latter half of the 20th century. Its journal, Kultura, played a key role in shaping the literary and intellectual minds of members of the Polish diaspora, while acting to many as the only source of Polish culture in the post-war era of Soviet censorship. It's book publishing efforts brought many new Polish authors to light in a world where otherwise they had no means to publish, and no audience to write to, actively courting contributions to periodicals and full publications from Polish authors around the world, and even from Poland itself

Today, *Instytut Literacki* has their publication history catalogued on their website. While it is very interesting to scroll through and can be used to look for specific articles and publications, in its current form it does not lend itself well to broader statistical analysis. The objective of this project was to create a dataset that could be analyzed on a macro scale, rather than observing the contents of individual texts, and to attempt to abstract some information from this data. Since I am not a data scientist this analysis is rather "proof-of-concept", and I defer the work of actually analyzing this information to those with more formal training and expertise. The hope is, however, that this data will help evaluate the degree to which *Instytut Literacki* achieved its goals of preserving the Polish diaspora's diversity and cultivating an atmosphere for debate among the Polish community, as well as understand what components make up its publishing catalogue.

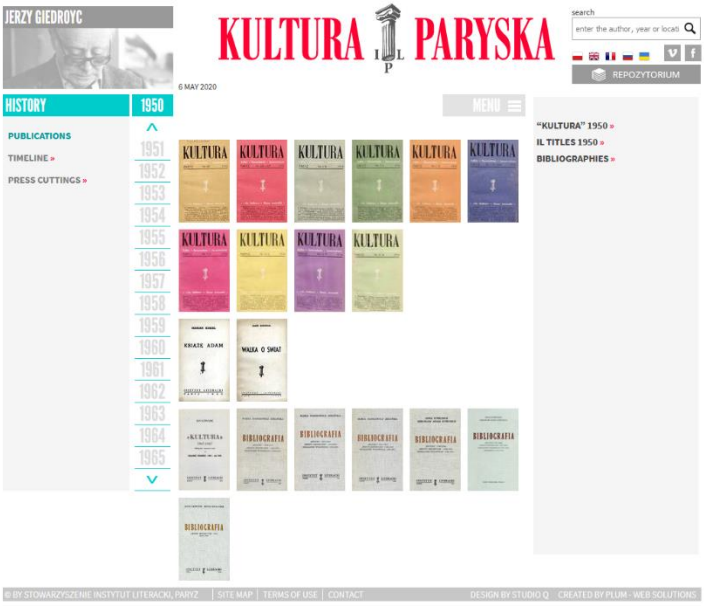
Web Scraper Design

The main idea of a web scrapper is to extract information from a webpage. One traditional application of this that demonstrate the "why" of web scraping would be to extract a list of products and prices from an online catalogue. If you wanted to build a tool that found the best deal on a certain product, you could build a web scraper that would search for a product on every marketplace you could think of, retrieve the price of the product, and then sort the marketplaces to find the best price. This would prevent you from having to check Amazon, Target, Best Buy, and Walmart, the next time you wanted a new appliance or TV.

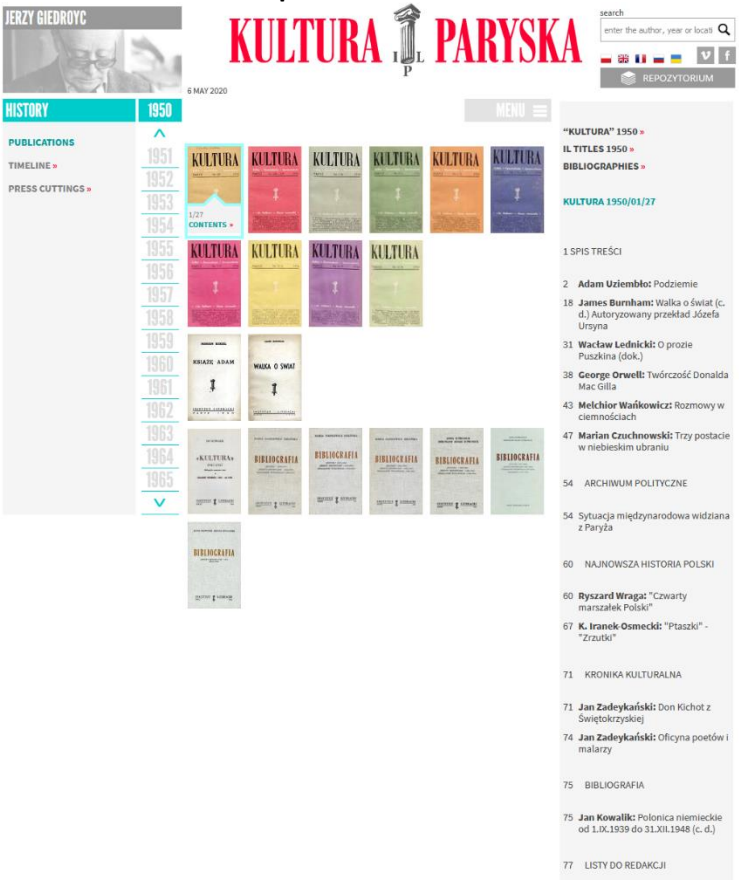
In this sense, a web scraper's main functionality is to retrieve preexisting data from a URL and collect or alter it in a way that provides some purpose or functionality. Often times, just to collecting data into one dataset can be useful for drawing conclusions about the data and enables many methods of statistical and data analysis. This is exactly the case we see with the *Instytut Literacki* publication catalogue.

To understand this scraper's functionality first we will have to look at the Kultura website. The list of publications is available here: http://kulturaparyska.com/en/historia/publikacje . Just as an example, clicking on the page for 1950 publications presents us with both periodical and book publications for the year:

This view is very nice for observing this information at a micro level. Clicking on each publication reveals, in the case of periodicals, a table of contents with attributions for each contribution as well as a link to view the contents of that edition.



List view of publications for 1950



Expanded view showing the contents of an individual edition

In this case we can easily observe specific details about one edition. But if you wanted to get an understanding of the larger picture across all editions of Kultura representing the information this way makes it practically impossible. If we were to extract this information to a table, we could actually look at larger trends in the data that is already here.

The views presented above are what this web scrapper will focus on. The main information to be extracted is in the case of periodicals: year of publication, edition number, list of articles with both author, and article title; in the case of books: year of publication, author, book title, long description of the book.

Year	Edition	Author	Article Title
1947	Kultura 1947/01	JÓZEF CZAPSKI	Raj utracony (na śmierć Bonnarda)
1947	Kultura 1947/01	TYMON TERLECKI	O socjaliźmie chrześcijańskim
1947	Kultura 1947/01	WIKTOR WEINTRAUB	Lytton Strachey
1947	Kultura 1947/01	ANDRZEJ BOBKOWSKI	Nekyia
1947	Kultura 1947/01	M.K. DZIEWANOWSKI	Wiosna Ludów w Hotelu Lambert
1947	Kultura 1947/01	ZYGMUNT ZAREMBA	Przeobrażenia wewnętrzne społeczeństw w okresie międzywojennym
1947	Kultura 1947/01	ARTHUR KOESTLER	Przyjazd (Z Krucjaty bez krzyża")"
1947	Kultura 1947/01	HERMINIA NAGLEROWA	W inne czasy (Fragment powieści)
1947	Kultura 1947/01	BENEDETTO CROCE	Zmierzch cywilizacji
1947	Kultura 1947/01	PAUL VALERY	Z Kryzysu ducha""
1947	Kultura 1947/01	TADEUSZ J. KROŃSKI1	Filozofia egzystencjalna Sartre'a
1947	Kultura 1947/01	FEDERICO G. LORCA	Wiersze
1947	Kultura 1947/02 - 03	Józef Czapski	Jangi Jul
1947	Kultura 1947/02 - 03	Józef Czapski	Dwa czasopisma
1947	Kultura 1947/02 - 03	Paweł Hostowiec	Recepty na przegranie wojny z Rosją

Small section representing the above data in a table

Representing data this way loses some of the detail of looking at the entire text. But striping away this information allows us to represent the publishing history of *Instytut Literacki* in other ways based on more scientific set of parameters.

My Code

All code can be found here: https://github.com/zdudzik/Kultura_Scraper including setup instructions and some preprocessed data files.

With an understanding of why applying a web scrapper addresses this problem, I will provide some brief insight into the architecture of this application. The flow of control goes through three states: the web scrapper retrieves information from the url; the data is categorized depending on its time of retrieval and it's underlying html tag, the data is stitched together in one of three data types and stored into a buffer that will later be converted into a csv formatted file. The decision to use csv is that it enables graphical based software such as Excel, Libre, or Cytoscape to read in and analyze the data, as well as being read again by either python or ruby to analyze via code.

Retrieval

Web scraping functions by looking at the html tree for tags on the elements of interest. The only way to decide what tags to search for is to look at html source code for a page, which in most browsers can be done by right clicking an element and selecting 'inspect element'. This reveals something like the html code fragment below:

```
<li style="padding-top: 10px;">
  <a target="_blank" href="http://static.kulturaparyska.com/attachments/8f/32/755bf61eceb808317f3dc79b67bc7a1a57529ce6.pdf#page=3">
    <span class="date2">2</span>
    <span class="title">
      <span class="nazwisko">Adam Uziembło:</span>
      <span class="tytul">Podziemie</span>
    </span>
  </a>
</li>
```

This code is taken from the html for the table of contents for an edition of Kultura. Here we can see two tags of interest containing the text “Adam Uziembło” and “Podziemiec”, which are the author and title of an article in this edition. In this case the span elements are given the classes “nazwisko” and “tytul”, which are searchable by the scrapper. Further investigation reveals that all articles on the webpage are formatted in this way, which means we can retrieve them all by looking for these two tags. For scraping, my code uses the ruby gem **nokogiri** which requires you only to provide the tag and class of interest. This segment of code contains the logic for searching for edition, author, and article title:

```
37          def parse_publication_data publication_link, contributions
38                  author_names = []
39                  article_names = []
40                  edition_name = ""
41
42                  #scrape title
43                  publication_link.css('h4.pub-title').each do |publication_data|
44                          edition_name = publication_data.content.strip.to_s
45                  end
46
47                  #scrape author name
48                  publication_link.css('span.nazwisko').each do |data_link|
49              author_names.push data_link.content.strip.tr(':','').to_s
50                  end
51
52                  #scrape article title
53                  publication_link.css('span.tytul').each do |data_link|
54              article_names.push data_link.content.strip.tr('\\','').to_s
55                  end
56
57                  #zip the lists together into individual entries, and add to contributions list
58                  author_names.length.times do |i|
59                          article = Contribution.new(author_names[i], article_names[i])
60                          contributions.push(article)
61                  end
62
63                  return edition_name
64          end
```

On line *48* you can see the reference to span.nazwisko, which we saw in the html tree

The scraping code can be found in the Controllers folder. Scraper.rb handles logic for periodicals, while Book_Scraper.rb handles book publications. Which logic to use is determined in main.rb via user input. The remainder of the information being scrapped is handled in a similar way to the one described above, and the logic can be seen in these files.

Categorizing

Once the data has been retrieved it needs to be categorized accordingly. There are three main models employed in this application and can all be found in the Models folder. For categorizing periodical publications there are three main components: year of publication, edition, and a list of contents, while each item in the content list has an author and a title. The file contribution.rb represents an individual article. The data retrieved from the nazwisko and tytul tags above get categorized into contribution objects. The file publication.rb defines a class that will hold the remaining information (edition, and year), as well as a list of contribution objects.

When completed each publication object will have:

- An edition number
- A publication year
- A list of contributions (all the articles)

and a contribution holds:

- Author
- Article Title

The final model is in book.rb, which defines the data to be stored for each book object:

- Author
- Book Title
- Text Description
- Publication year

Output

Output is handled after all data has been retrieved and the data objects. Contained in each model class is a method for outputting to csv. Each book or article retrieved is printed line by line into a file following csv formatting.

Data Visualizations & Conclusions

Data files can be generated using the code but are also provided already exported into excel for ease of use. The file *kulturea_data_all_years.xlsx* is an excel sheet containing the scrapped information for all periodical publications. The pivot table included is an example of the types of analysis that can be done with this data.

While more in depth analysis can be done, the data presented here will focus on author publication histories with respect to both book and article publications. Below are graphs illustrating the volume of work published in each year by the most published authors. Higher resolutions are available in the excel files in the repository.

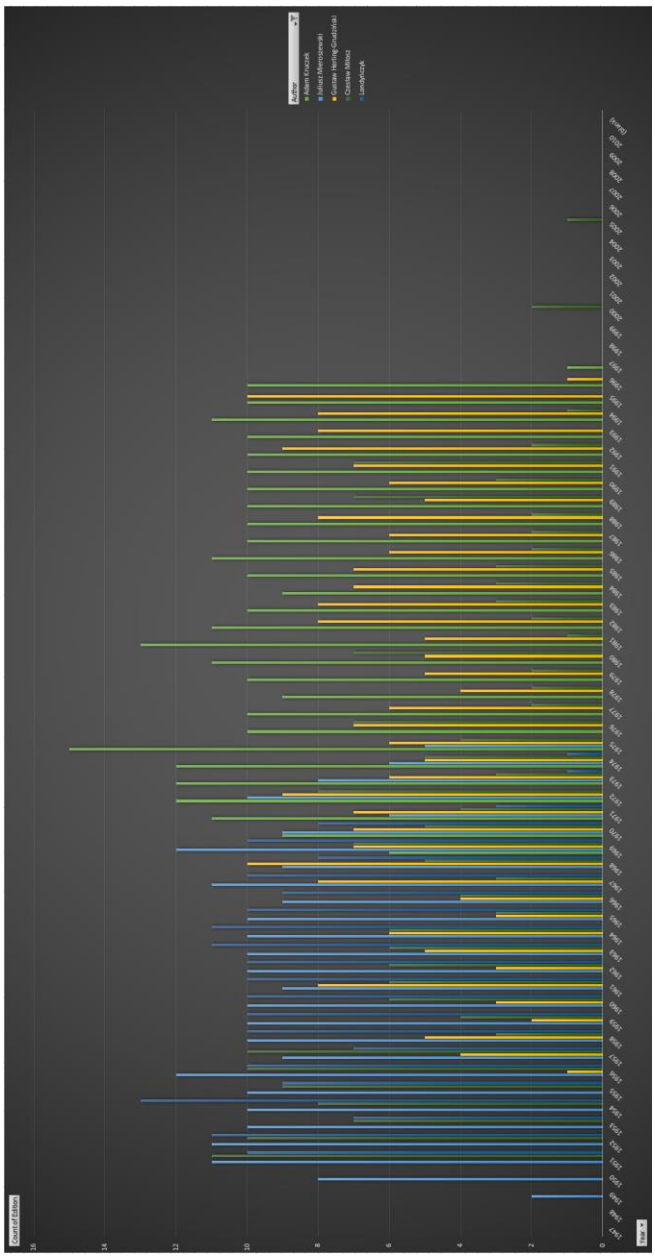


Figure 1: Filtered for top 5 authors. Represents number of periodical publications per year by each author

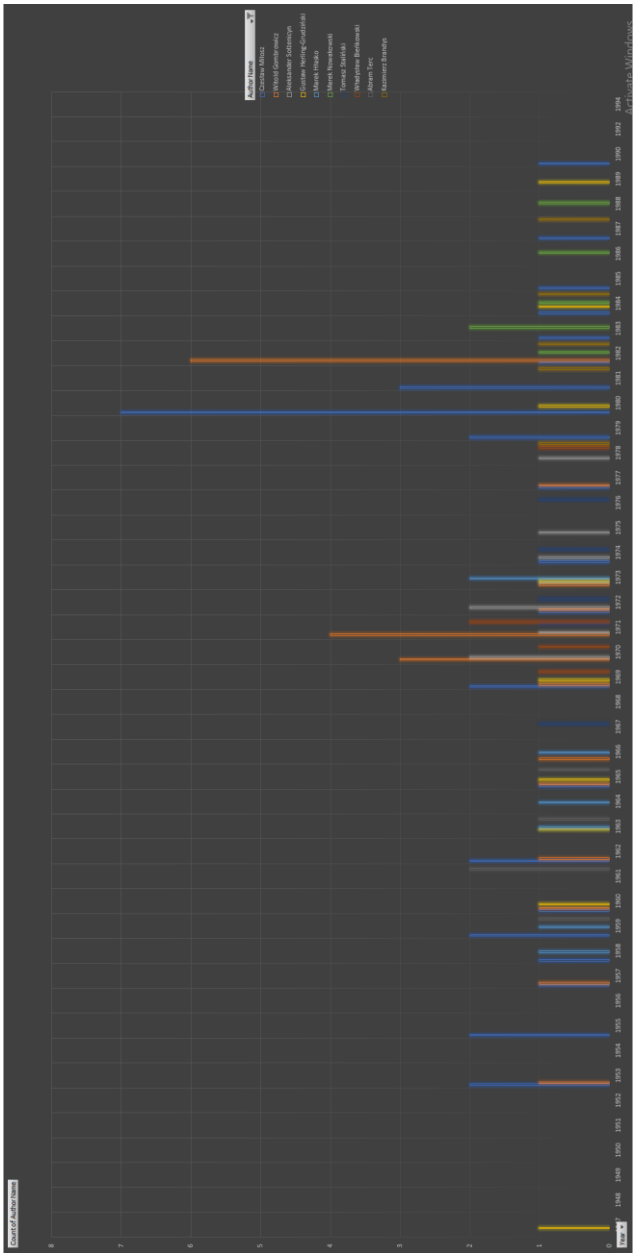


Figure 2: Filtered for top 10 authors. Represents number of book publications per year by each author

These figures can reveal a number of things, but the data for the periodical publications in this context is a bit more interesting, in part to the larger amount of data. First here are some basic facts about the data:

> From 1946 to 2010 there have been 808 periodicals published (637 issues of Kultura and 171 issues of Historical Notebooks). These 808 editions have contained 14,544 different articles (averaging 18 articles per edition) by 4,193 different contributors. 55 individuals account for over 33% of contributions to Kultura or the Historical Notebooks, and just 7 individuals account for more than 10% of contributions. The 5 biggest contributors where Adam Kruczek (with 293 articles), Juliusz Mieroszewski (247), Gustaw Herling-Grudziński (245), Czesław Miłosz (214), and Londyńczyk (199).
>
> From 1947 to 1994 there have been 385 books published by 198 different authors. The most published authors are: Czesław Miłosz (35 publications), Witold Gombrowicz (23), Aleksander Solzhenitsyn (9), Gustaw Herling-Grudziński (9), and Marek Hłasko (7).

There are a few ways to interpret this data. When evaluating *Instytut Literacki's* goal of representing a wide range of polish literature one may point to the large number of contributors (4,193) and authors (193) and say that indeed yes, they succeeded in representing a varied Polish voice. However, one may also argue that 20 authors contributed over a quarter of articles featured in periodicals published, and many of these authors either lived and worked in Paris or had very close ties with the institute and with each other. In this respect, *Instytut Literacki* does not represent a wide range of authors well, instead relying heavily on the contributions of a much smaller inner circle.

Also interesting to note, is the influence of specific figures over certain periods. Looking at Figure 1 above reveals two regions. Before 1970 almost all of the article contributions are done by Julius Mieroszewski and Londyńczyk (perhaps a pseudonym meaning Londoner). However, after Mieroszewski's death in 1974, the majority of contributions are now from Adam Kruczek and Gustaw Herling-Grudziński. It would be interesting to do a micro analysis on texts from these specific periods to see if there are major thematic differences between these two "eras".

There are other questions left unanalyzed that would be interesting to look at but are ultimately out of my realm of expertise at the moment. Primarily: during which time periods were contributions more diverse? Do a large number of one-time contributors appear during certain periods? This could reveal:

- Were there times that the institute relied heavily on a few contributors. This could indicate times of particular difficulty with respect to censorship. How were these editions received by readers? How were more diverse editions received, were they less popular?

- What are the underlying causes that lead to a “one-time” contributor? Is courting the contribution too difficult? Is the large number of one-time contributors a testament to the institute achieving its goal of diversity, or simply a biproduct of censorship laws?

It would also be worth researching more heavily the texts of the few writers who’s works made up such a large portion of publication. These writers likely had a much larger impact on the views of the readership, and it is worth analyzing:

- What did these writers write about? What are the themes of their texts, what are their backstories?
- What made these writers more popular? Was it their pure skill, or did they simply have easier access to print than other writers?

These are just a few questions that can be answered looking at data through this lens, questions that provide a different kind of understanding to this publishing house, and its émigré community. Perhaps the real strength of this technique is that it can provide strong guidance on what other research may be done in the future, by showing us what questions we can ask.