University of Warsaw

Faculty of Economic Sciences

Maciej Sacharczuk

Nr albumu: 410854

Michał Sękowski

Nr albumu: 411399

Michał Staniaszek

Nr albumu: 411476

Gathering Roll-Call data from Senate of Poland website using web-scraping techniques

Final Project prepared

for Webscraping course taught by
Maciej Wysocki and
Przemysław Kurek

Warsaw, May 2022

Project description

Our project focuses on gathering Roll-Call type of data from Senate of Poland website (https://www.senat.gov.pl/). We were particularly interested in data from 9th cadency, as it is the newest one that has already ended. Roll-Call data has been proven as very valuable in Political Sciences and it is commonly used in analyzing American political scene. Well-known example of its usage is measuring changes in polarization among House of Representatives members and ilustrating worldview of those by points in 2-dimensional space. Data gathered by spiders created in this project allow for the same kind of analysis but for Polish political scene.

Scraper mechanics and output data

All 3 of our scrapers utilize same mechanic:

Open

https://www.senat.gov.pl/prace/posiedzenia/tematy-posiedzen-senatu-ix-kadencji/

as a starting page. This site contains all the links that lead to websites dedicated to specific sessions during 9th term of Senate. There have been 85 sessions, so our spiders gather list of 85 links on this first page.

On each of those 85 sites from previous step (here is example link to the first one of them: https://www.senat.gov.pl/prace/posiedzenia/tematy,444,1.html) get the link under "Głosowania" button. After doing it for all sites, we end up with list of 85 links of sites containing details of each sessions, such as votings that took place then. First link from this list looks like this:
https://www.senat.gov.pl/prace/posiedzenia/przebieg,444,1,glosowania.html

We noticed that those links are quite similar and that this pattern repeats throughout all links in the list. Therefore instead of having our spiders enter links from the first list just to gather one link from each, we did some string replacements and thus completely omitting this step. That is, for each link out of 85 links from the first list:

We replace string"tematy" with "przebieg"

We replace string"1." With "1,glosowania"

And we end up with the same list of links. Now, on each of those sites, we want to scrap links leading to final roll-call data. We are interested in printable version of those, so we will need links under "wersja do druku" button. We have to account that there are some sessions where are multiple Roll-call votings, and some where there isnt one.

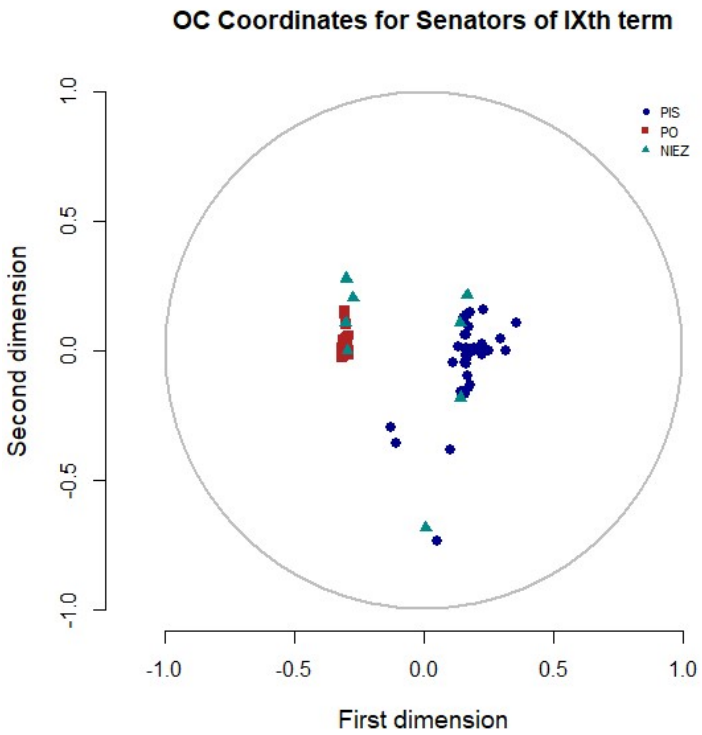
After completing this step, we have 517 links directing to roll-call data of 517 votings. Final step is to gather roll-call data from those pages. From each of these sites, we collect data of all senators that did vote then and also how did they vote. Then, next to each record we also put link to site from which that record was received. This helps in further analysis, as those links are unique for each voting and number inside that link is growing with next votings what makes putting those roll-calls in order easy.

Final result is complete Roll-Call dataset, containing 60595 records, each of those consisting of Senator full name, how did he vote (possible values: “za”-“yea”,”przec.”-“nay”,”nie gł.”-“didn’t vote”,”wstrz.”-“abstained”) and link toward specific voting data. As requested all of our scrapers contain 100-page limit switch. When its turned on, there are 9765 records. Below is tiny snippet of obtained dataset:

W. Bonkowski	nie gł.	https://www.senat.gov.pl/prace/posiedzenia/glosowanie-drukuj,8274.html
M. Augustyn	wstrz.	https://www.senat.gov.pl/prace/posiedzenia/glosowanie-drukuj,8608.html
G. Bierecki	za	https://www.senat.gov.pl/prace/posiedzenia/glosowanie-drukuj,8608.html
P. Błaszczyk	za	https://www.senat.gov.pl/prace/posiedzenia/glosowanie-drukuj,8608.html
A. Bobko	za	https://www.senat.gov.pl/prace/posiedzenia/glosowanie-drukuj,8608.html
M. Augustyn	przec.	https://www.senat.gov.pl/prace/posiedzenia/glosowanie-drukuj,8607.html

Analysis:

After some simple transformation of dataset and assigning each senator to political party he represented during 9th cadency of Senate of Poland, we were able to input Roll-call data to multidimensional scaling tool called Optimal Classification. While its extremely advanced software, fortunately it works in black-box manner. What it basically does is producing map of points, where each point represents worldview that one of senators revealed by voting behaviour throughout cadency. Generally speaking, if we consider pair of points, the closer they are to each other, two of senators that are represented by them voted in more similar way.



Blue points are represented by Law and Justice party (PiS), Reds are Civic Platform (PO) while green are indepentens. As we can see, there is very high discipline among PO members, which back then had majority in Senate. Points representing them are extremely close to each other, meaning that rarely any one of them voted differently than the rest. Members of Law and Justice discipline isnt as high. Those senators that are far away from both groups, are usually senators that switched party mid-term, and in more complex analysis are treated differently.
We believe that this short although exciting analysis show the power of gathered data.

Performance comparison

In order to test speed of 3 scrapers, we run them with limit boolean set to False (scraping without 100-page limitation). Scrapy was the fastest with sub 3 minute time, but Beautiful Soup took around 3 minutes as well. The slowest one was Selenium scraper, which took roughly 30 minutes (this is partially due to 1-second delays set in code).

Work partition:

Maciej Sacharczuk – Beautiful Soup webscraper, Description document

Michał Sękowski – Scrapy webscraper, Analysis, Github setup

Michał Staniaszek – Selenium webscraper, Description document