Crawler For Google Scholar

DataMining

Base Idea

Give a url of a certain author, like *Jie Tang*'s Google Scholar page: http://scholar.google.com.sg/citations?user=n1zDCkQAAAAJ&hl=zh-CN.

We can easily crawl the basic information of the author and the list of the author. I notice that there is a list of co-authors of the author. Then we can use these urls to operate the bfs(board first search) and crawl the whole co-author network.

Implementation

- I use the *Scrapy* Framework in **Python** to implement the crawler. Right now, I just save my results in the format of *json*.
- To filter the duplicate page crawled, I use the *set* object in **Python** to save all the url crawled and crawl new pages after checking if it's in the *set*.
- In order to prevent the blockage of Google, I write several specific crawlers to crawl thousands free proxy IP and port on some websites: www.proxy360.cn, www.cnproxy.com. And use these proxies in a certain way to hit the target pages and then avoid the detection of Google.
- To improve the efficiency, I also filter these proxies to obtain the effective proxies with my IP (I use VPN with the IP in Hong Kong). And in different environment and time, I can reuse my code to get different effective set of proxies to achieve better performance of the crawler.
- We can also change the **DOWNLOAD_DELAY** to avoid hitting the servers too hard and the dection of Google with the expense of efficiency to some degree.

Result

I use the start url of *Jie Tang*'s page and the crawler can crawl a network of 1000 authors and about 15000 papers within one hour without blockage.

In ideal situation, the crawler can crawl up to 20000 authors and 300000 papers in a day.

However, it still needs to be tested further.

Information Crawled

For a certain author, the information I crawled:

- URL
- Name
- Information (Postion and Organization)
- Paper List
    - Name
    - Author List
    - Journals or Conferences
    - Citation Number
    - Year of Publication

Further Work

- Write the pipeline to save the data to some database.
- Now, I just can obtain part of the paper due to the page layout, just as the pic below.

  And maybe I should try some methods to get the whole list.

Appendix

All my code can be found on the Repo on GitLab.

Prerequsites

- Scrapy