Web Crawler

وب کرالر نوشته شده به این صورت عمل می کند که در ابتدا یک عدد به عنوان ورودی برای تعداد مقاله های حدودی برای واکشی از سایت می گیرد.
fetched_articles دیکشنری است که در طول زمان مقاله های واکشی شده در ان (ایدی مقاله به عنوان کلید و داده های واکشی شده به عنوان مقدار) ذخیره می شوند.

سپس می توان مشخص کرد که از چند باروزر به صورت موازی استفاده می کند.از آنجایی که سرور های سایت محدودیتی در تعداد درخواست در زمان مشخصی دارند با اضافه کردن تعداد بیشتر باروزر سرعت واکشی حتما بالا نمی رود اما می توان با ست کردن تعداد باروزر به ۲ یا ۳ زمان پردازشی لود شدن صفحه و ارسال درخواست را کمتر کرد، که در اینجا این مقدار ۳ تنظیم شده و داده های ذخیره شده در فایل با ۳ باروزر واکشی شده اند.
در صورتی که تعداد بارزر ها زیاد باشد ممکن است در زمان های اول کار صف مقاله ها برای واکشی برای مدتی خالی شود در این صورت باروزر ها باید صبر کنند تا بقیه باروزر ها درخواست واکشی را کامل کنند تا در درون صف مقاله های جدیدی گذاشته شود برای همین متغییری از لیست بولین ها تعریف شده به نام browsers_empty_q_flags به این صورت که هر باروزر اگه به صف خالی برخورد کرد تا زمانی صف پر نشده مقدار بولین مربوط به خود را در این آرایه به True تغییر می دهد که بقیه باروزر ها بفهمند صف خالی است و درصورتی که همه باروزر ها به صف خالی برخورد کنند یعنی اینکه مقاله ای قرار نیست به صف اضافه شود و برنامه تمام شود.
در طول برنامه ترد ها مقاله هایی که می خواهند واکشی کنند را اول چک می کنند که حتما قبلا واکشی نشده باشند و سپس اقدام به واکشی می کنند.

برای واکشی هر مقاله برای هر باروزر تابع fetch_article_data_with_link تعریف شده که از باروزر مورد نظر اطلاعات مقاله مورد نظر را واکشی می کند.

توضیحات مربوط به تابع fetch_article_data_with_link

برای این تابع دو ورودی browser , link تعریف شده است.
برای واکشی داده ها از کتابخونه selenium استفاده شده است زیرا بسیاری از اطلاعات مقاله ها بعد از لود شدن صفحه و با Ajax در صفحه قرار می گیرند برای همین این امکان وجود نداشت به یک Get ساده اطلاعات را بگیریم.از انجایی که نمی دانیم چه مدت طول می کشد که اطلاعات مورد نیاز ما با Ajax لود شوند تابع execute_until_done .
execute_until_done به عنوان ورودی یک تابع می گیرد و تا زمانی که این تابع را بدون خطا کال نکرده است صبر می کند. این تابع تلاش خود را برای کال کردن تا مدت ۴ ثانیه انجام می دهد و در صورت موفق نبودن یه خطا ایجاد می کند که در این صورت برنامه این لینک را در به ته صف منتقل می کند و سرغ لینک بعدی می رود.
تابع دیگری به نام wait_until_load نیز تعریف شده که قبل از واکشی داده ها از سایت صدا زده می شود و تا زمانی دستور "return document.readyState" مقدار complete را برنگردانده است، صبر می کند.

هر کدام از اطلاعات مقاله به صورت زیر استخراج می شوند:

id:
link.split("/")[-1]

title:
execute_until_done(lambda: browser.find_element_by_class_name("name").text)

abstract:
execute_until_done(lambda: browser.find_elements_by_tag_name("p")[2].text)

year:
execute_until_done(lambda: browser.find_element_by_class_name("year").text)

all_authors_tag:
browser.find_elements_by_xpath("//*[@data-appinsights-key.bind='author.id']")

ref_count:
int(execute_until_done(lambda: browser.find_element_by_class_name("count").text))

و درصورتی که تعداد مقالات نوشته شده در بالای صفحه(ref_count) بیشتر از صفر تا باشند اقدام به گرفتن خود رفرنس های می کند زیرا فرآیند پیدا کردن رفرنس مقداری زمان بر است با این کار سرعت کار افزایش می یابد:

all_references_tag:
browser.find_elements_by_class_name("primary_paper")