

一、工具与环境

使用 Anaconda 虚拟环境，安装 python3、pyquery、scrapy

```
pip install pyquery
pip install scrapy
```

暂未使用 selenium

浏览器使用 Microsoft Edge 123.0.2420.65 (正式版本) (64 位)

二、如何爬取

1. scrapy 与 wiki

1.1 wiki

scrapy 提供了较为完整的框架，在爬取 wiki 数据的时候使用


scrapy 基础使用可参照 [Scrapy 入门教程 | 菜鸟教程 \(runoob.com\)](#)，更多内容在 [Scrapy 2.11 documentation — Scrapy 2.11.1 documentation](#)

在对项目进行初始化后生成一个自己的类，设置其中的 allowed_domains

```
allowed_domains = ['zh.wikipedia.org']
```

维基有一个单独的搜索页面，不同于普通搜索框在输入关键词后会直接跳转到对应词条，这个页面可以展示所有与关键词相关的词条，在搜索框内无内容时点击“搜索”即可

≡



维基百科
自由的百科全书

Q 搜索维基百科

搜索

维基百科Discord、IRC、LINE、QQ及Telegram等各平台交流群欢迎大家加入。

搜索

Q

搜索

高级搜索：

排序依据：

搜索：

条目 ×

此时查看域名变为 `https://zh.wikipedia.org/w/index.php?search=&title=Special:搜索`

`search=` 后即为要搜索的关键词

1.2 举例：搜索总统选举

```
url = https://zh.wikipedia.org/w/index.php?title=Special%3A搜索
&limit=50&offset=0&ns0=1&search=总统选举
```

`limit` 属性为一次查询的条目数量，`offset` 为默认查询结果的偏移量（从第几个开始）

有了这个 `url` 后就可以开始编写爬虫

scrapy 类中的 `start_urls` 为爬取的起始列表，可以将此 `url` 放入

```
start_urls = ("https://zh.wikipedia.org/w/index.php?title=Special%3A%E6%90%9C%E7%B4%A2&limit=50&offset=0&ns0=1&search=%E6%80%BB%E7%BB%9F%E9%80%89%E4%B8%BE",)
```

`%E6%80%BB%E7%BB%9F%E9%80%89%E4%B8%BE` 为粘贴后的编码问题，不影响使用

`parse` 方法拿到的 `response` 数据为网页源代码，在维基页面按 `F12` 即可查看：



维基返回的搜索数据全部放在一个 `ul` 里，每个 `li` 都对应一个条目

在下面的 `a` 标签中，这个链接就是我们想要的详情页链接：

```
<li class="mw-search-result mw-search-result-ns-0">
  <table class="searchResultImage">
    <tbody>
      <tr>
        <td class="searchResultImage-thumbnail">
          <td class="searchResultImage-text">
            <div class="mw-search-result-heading">
              <a href="/wiki/1824%E6%90%9C%E7%B4%A2&limit=50&offset=0&ns0=1&search=%E6%80%BB%E7%BB%9F%E9%80%89%E4%B8%BE" title="1824年美国总统选举" data-serp-pos="0">
                </div>
              <div class="searchresult">
                <div class="mw-search-result-data">43 KB (5,633个字) - 2023年9月19日 (二) 10:50</div>
              </div>
            </td>
          </tr>
        </tbody>
      </table>
    </li>
```

用 `xpath` 解析网页内容可以得到所有的目标 `url`，这里将其以 `link,title` 的格式写入文件（注：爬取的 `url` 为相对路径）

```
for each in response.xpath("//div[@class='mw-search-result-heading']"):
    link = each.xpath("a/@href").extract()
    title = each.xpath("a/@title").extract()
    file.write("https://zh.wikipedia.org" + link[0] + ',' + title[0] + '\n')
```

现在这个文件 `a.csv` 中是关于“总统选举”的所有目标网址和标题，再使用一个爬虫依次获取网页内容

由于只获取文本信息，可以再看看维基详情页的结构：



所有文本都包含在 `p` 标签下，使用 `response.xpath("//p//text()")` 便可拿到所有文本。如果想同时拿到标题，增加 `//h2//text()` 等等

直接获取的维基数据简体繁体混杂，推荐使用 `opencc` 进行繁转简：[BYVoid/OpenCC: Conversion between Traditional and Simplified Chinese \(github.com\)](https://github.com/BYVoid/OpenCC)

最后以标题作为文件名，全部 `p` 标签下的文本作为内容输出即可

2. pyquery 与新华网

`scrapy` 同样能胜任这份工作，只不过在使用中发现 `pyquery + request` 更简单

有关 `pyquery` 的简单使用，可以参考[Python pyquery 教程 | 极客教程 \(geek-docs.com\)](https://www.geek-docs.com/python/pyquery/)，全部内容在[pyquery - PyQuery complete API — pyquery 2.0.x documentation](https://pyquery.org/)

`pyquery` 是用来解析网页的工具，是 `jquery` 的 `python` 实现，网页的获取需要使用 `request` 库

2.1 新华网

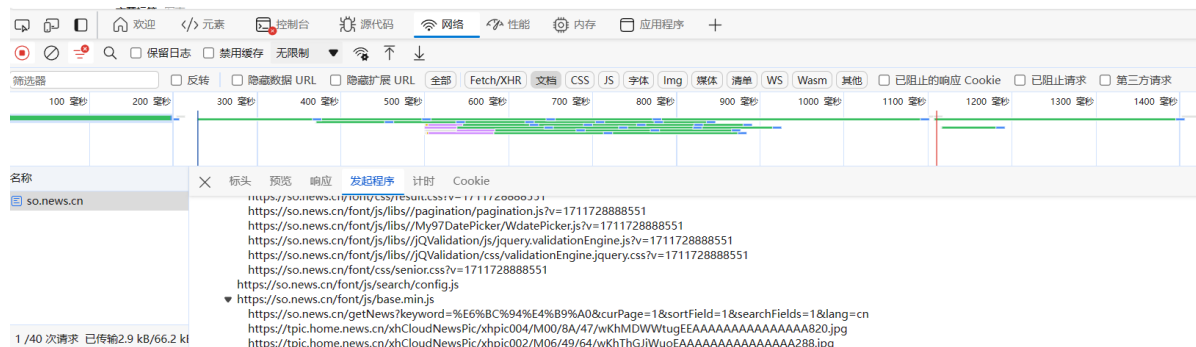
新华网的好处：大量新闻，搜索功能不错，大部分新闻第一段内容可作为结果，不需要动态加载（处理简单）

新华网搜索域名：`http://so.news.cn`

2.2 举例：搜索演习

新华网无法直接从 `url` 获取到网页数据，直接爬取 `https://so.news.cn/#search/0/演习/1/` 会发现什么都没有

按 `F12` 打开控制台，`ctrl + R` 刷新网页，在 `Fetch/XHR` 或 `文档-发起程序` 下可以找到真正的请求：



取出标题和内容也很简单：

```
for item in source('.title').items():
    title = item.text()
    break
for item in source('#detail p').items():
    content = content + item.text() + '\n'
    #break
```

对于上面代码中的第二个 `break`，由于**大部分**新闻在第一个 `p` 标签下会用很简洁的话做为概括，所以只需要概括的时候可以取消其注释

当然这样写会带来一些问题，比如在第一段话前有空的 `p` 标签，或是新华网链接到外部网站导致 `(.title)` 和 `(#detail p)` 什么都取不到。由于新闻数量足够多，这些暂时是可以忽略的

另：一些事件数据源

1. GDELT 数据库

[The GDELT Project](#)

同个文件夹下的 `csv` 文件即为一份 `GDELT` 示例

`GDELT` 是一个全面记录事件和事件参与者的时空数据集，其核心是对事件及其参与者信息的自动化识别、概化、分类和编码。2013 年 `GDELT` 数据库公布的第一个版本包含自 1979 年以后的所有事件，更新频率为 1 天。2015 年 `GDELT` 的第二个版本更新频率提高到了 15 分钟，但时间范围仅从 2015 年 2 月 19 日开始。截止 2018 年底，`GDELT` 记录的事件总数超过 7 亿 条。每条 `GDELT` 数据记录主要由 5 个部分组成，分别是事件编号和日期、事件参与者、事件动作、事件地理信息以及数据管理。

`GDELT` 的数据格式为包含活动参与者、国家和地区信息、组织形式及各种代码在内的几十列处理过的信息，想得到原文本可以查看 `SOURCE_URL` 字段提供的网址——它的主要缺点也在这里：源网址是多语言的，可以用来训练的中文文本很少。

官网提供的查询方式现在已失效，提供的发邮件查询方式也不可行，除此之外还有收费非常高的 `Google BigQuery` 服务。所以如果不要求事件类型可以使用官网提供的压缩包下载获取事件，这些事件只按照发生时间排列，否则没有很好的方法。

北师大

北师大似乎搞了一个以 `GDELT` 为数据源的项目，但并未尝试过爬取这个网站。

[全球新闻事件数据共享平台 \(bnu.edu.cn\)](http://bnu.edu.cn)

2. ICEWS 数据库

[Integrated Crisis Early Warning System | Lockheed Martin](#)

文件形式类似于 GDELT，缺点还是无中文语料

3. 百度 DuEE、DuIE 数据集

[DuEE1.0中文事件抽取数据集](#)

[DuEE-fin金融领域篇章级事件抽取数据集](#)

[DuIE2.0中文关系抽取数据集](#)

中文事件库，分别有 11w 和 17w 数据量；缺点是事件通常只有一句话，且涵盖类型过多