Beijing Forest Studio

北京理工大学信息系统及安全对抗实验中心

爬虫中的攻与防

韩飞　硕士研究生

2020年05月10日

内容提要

- 背景简介
- 基本概念
- 算法原理
- 优劣分析
- 应用总结
- 参考文献

背景简介

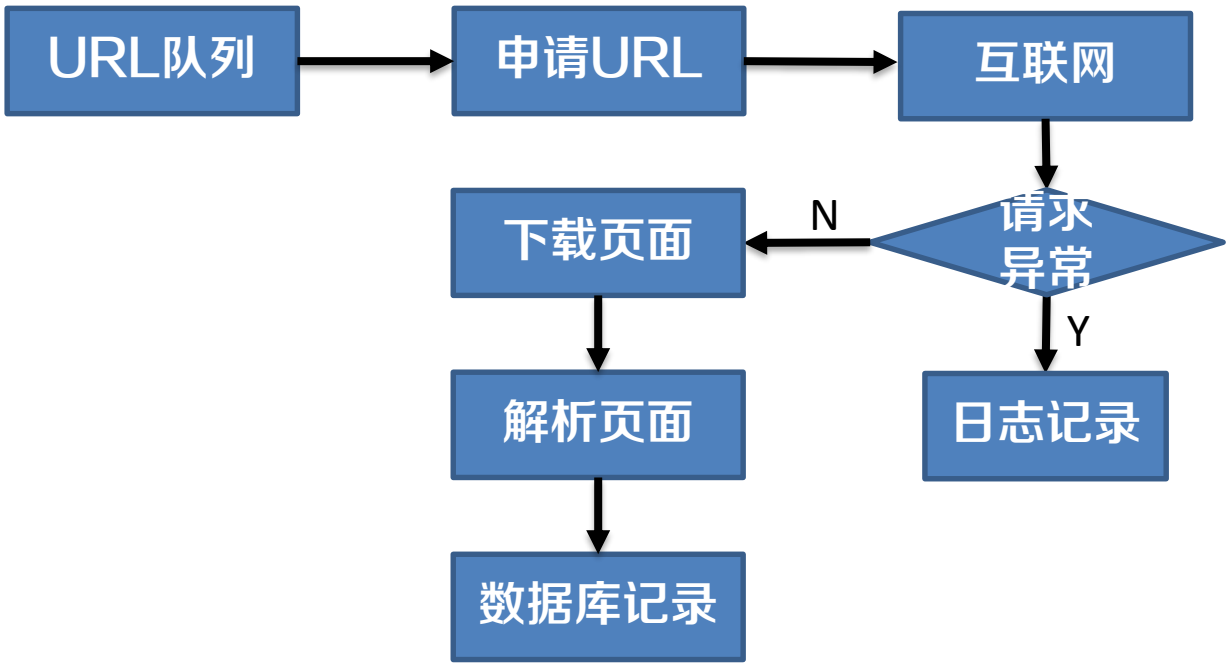
- 网站对抗爬虫使用的反爬手段
- IP限制，短时访问限制，UA识别限制，动态加载……
- 爬虫应用中应对反爬场景的对抗手段
- 设置IP代理池，随机数延迟，header伪造，模拟浏览器driver获取页面……

背景简介

- 预期收获
- 了解网站中使用比较频繁的反爬手段
- 了解网络反爬手段的常见对抗措施

基本概念

- 爬虫



基本概念

- http
- http是一个简单的请求-响应协议，它通常运行在TCP之上。它指定了客户端可能发送给服务器什么样的消息以及得到什么样的响应。请求和响应消息的头以ASCII码形式给出；而消息内容则具有一个类似MIME的格式。

- Header
- 在HTTP的请求和响应消息中，协议头部分的那些组件。HTTP消息头用来准确描述正在获取的资源、服务器或者客户端的行为，定义了HTTP事务中的具体操作参数。

基本概念

- User-Agent　浏览器的身份标识字符串
- Cookie　由之前服务器通过Set-Cookie设置的一个HTTP协议Cookie

- Ajax
- Asynchronous JavaScript and XML，是一种在无需重新加载整个网页的情况下，能够更新部分网页的技术。

基本概念

- html的
<iframe>

-
 可插入一个简单的换行符。
-
 标签是空标签（意味着它没有结束标签，因此这是错误的：
</br>）。在 XHTML 中，把结束标签放在开始标签中，也就是

- <iframe>
- iframe 元素会创建包含另外一个文档的内联框架（即行内框架）。

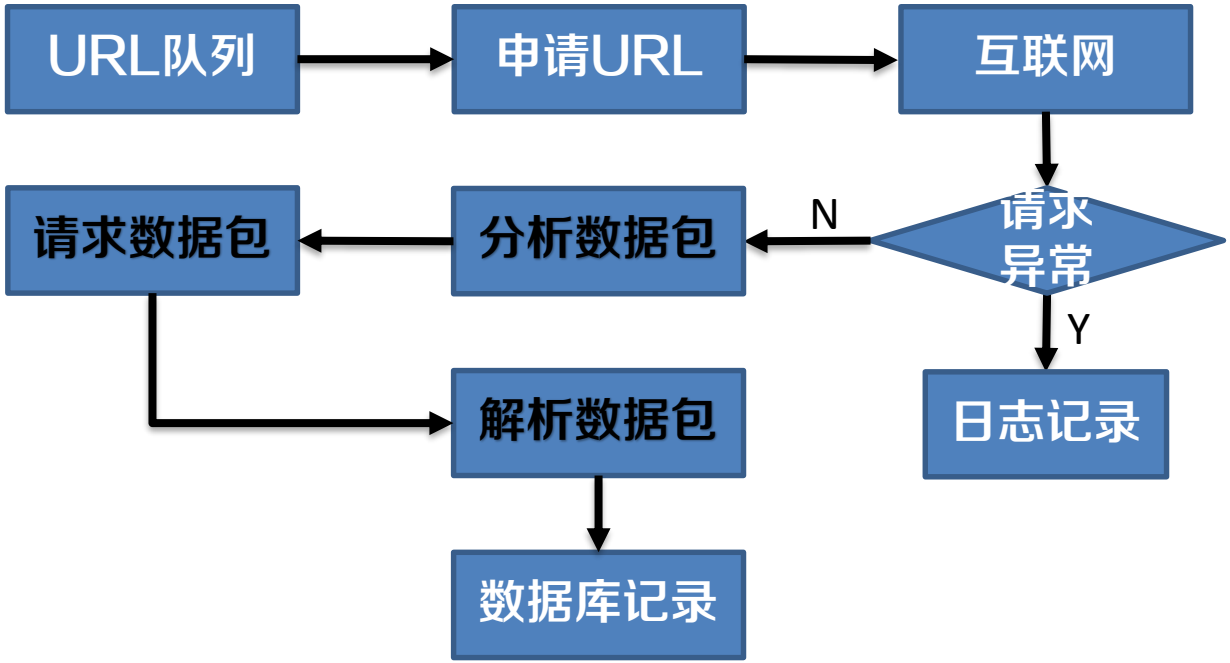
算法原理

- 1. 直接抓取http请求中的json数据包对抗网站设置的动态页面加载

T	获取ajax加载或者javascript渲染的网页中的文本信息
I	ajax加载或者javascript渲染的网页url
P	分析ajax加载或者javascript渲染的子网页的数据包地址
O	ajax加载或者javascript渲染的网页中的文本信息

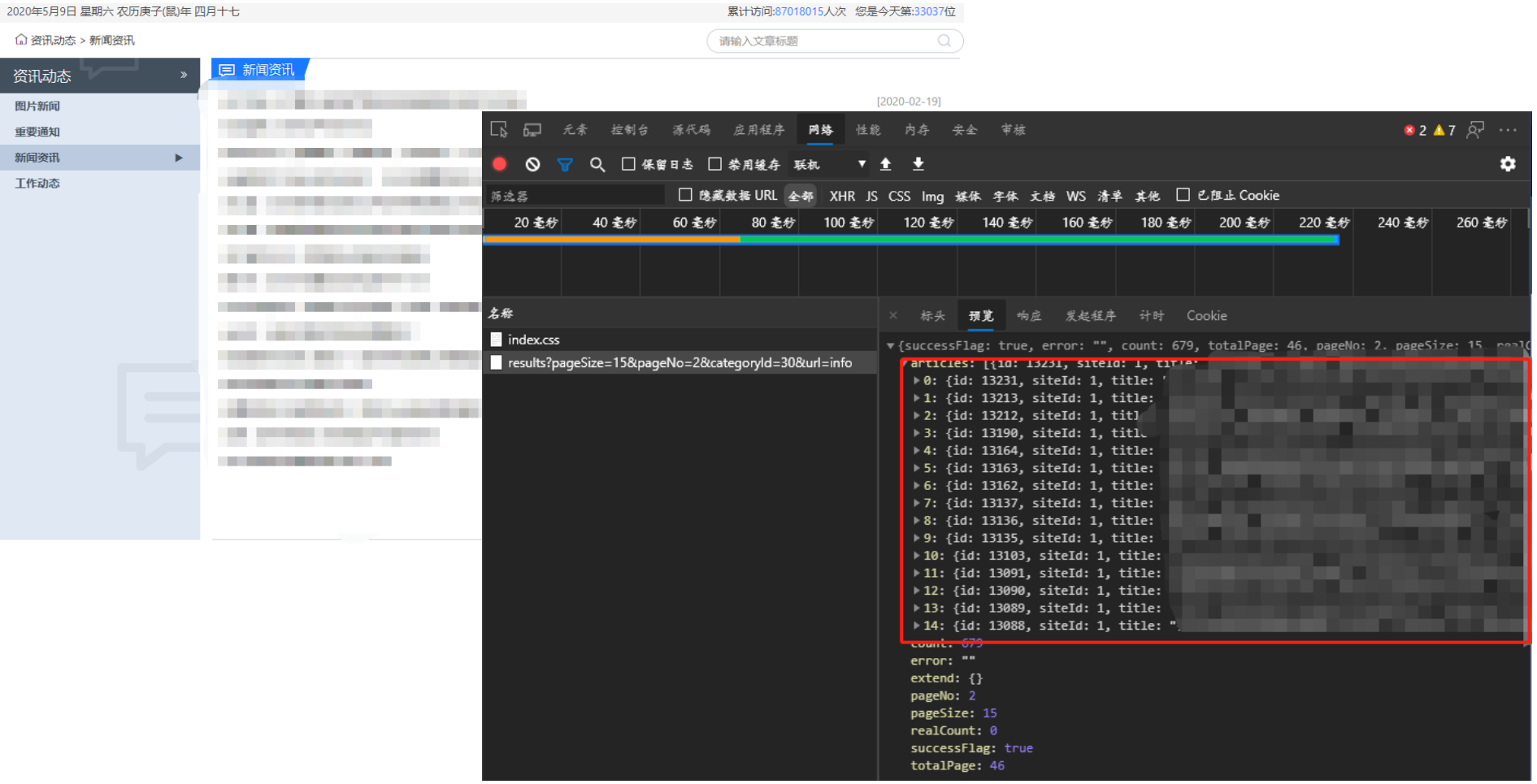
算法原理

- 流程图



算法原理

- 场景，以某省新闻网站为例



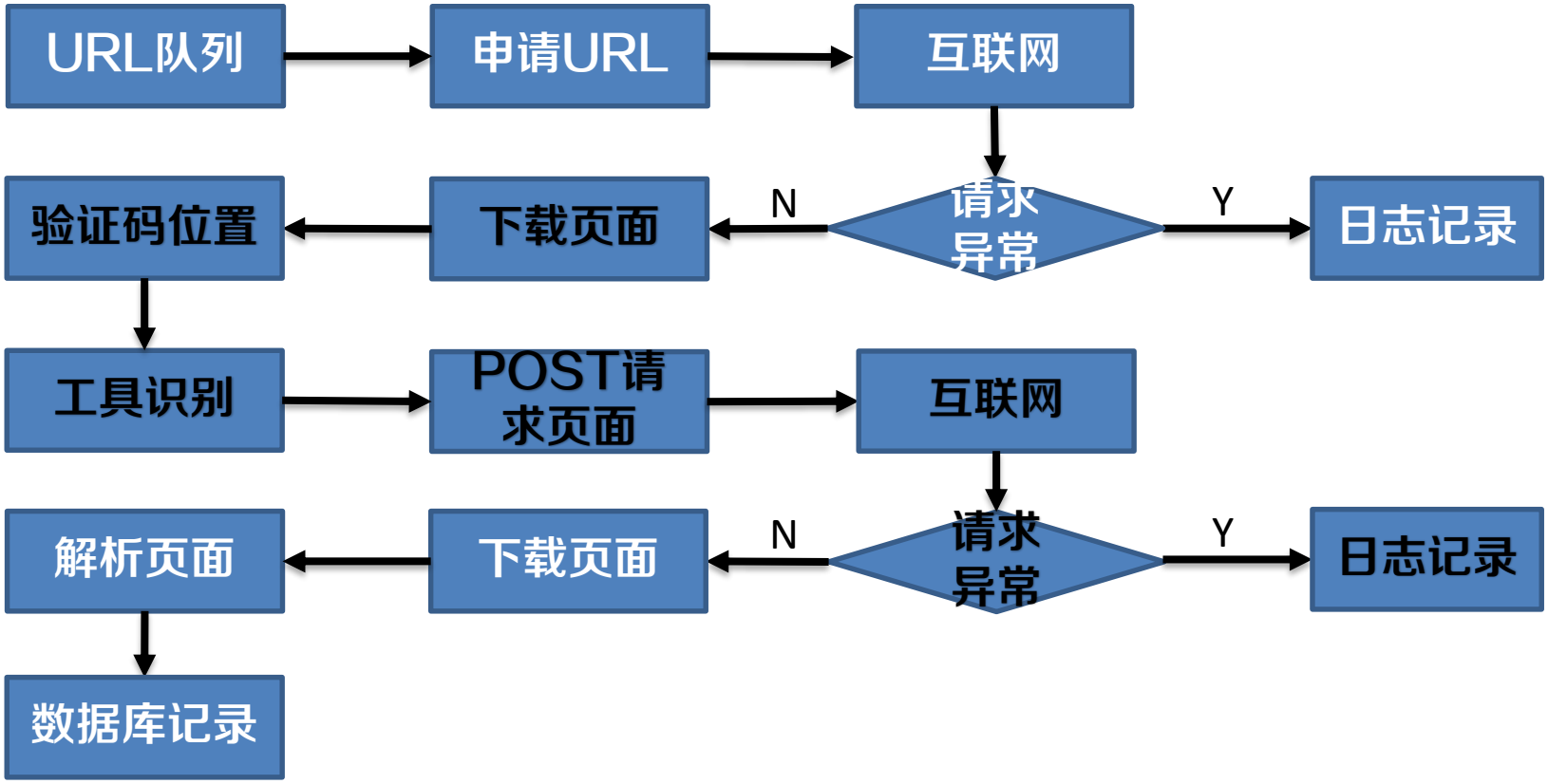
算法原理

- 2. 将请求网页遇到的验证码使用网络上的打码平台、python图像识别库以及词表库自动填写；需要登录验证的网页通过伪造能够成功访问的session/token达到对抗反爬目的

T	获取需要验证码/登录验证的网页文本信息
I	需要验证码/登录验证的网页url
P	将验证码下载并使用网络上的打码平台、python图像识别库以及词表库自动填写； 伪造能够成功访问的session/token
O	需要验证码/登录验证的网页文本信息

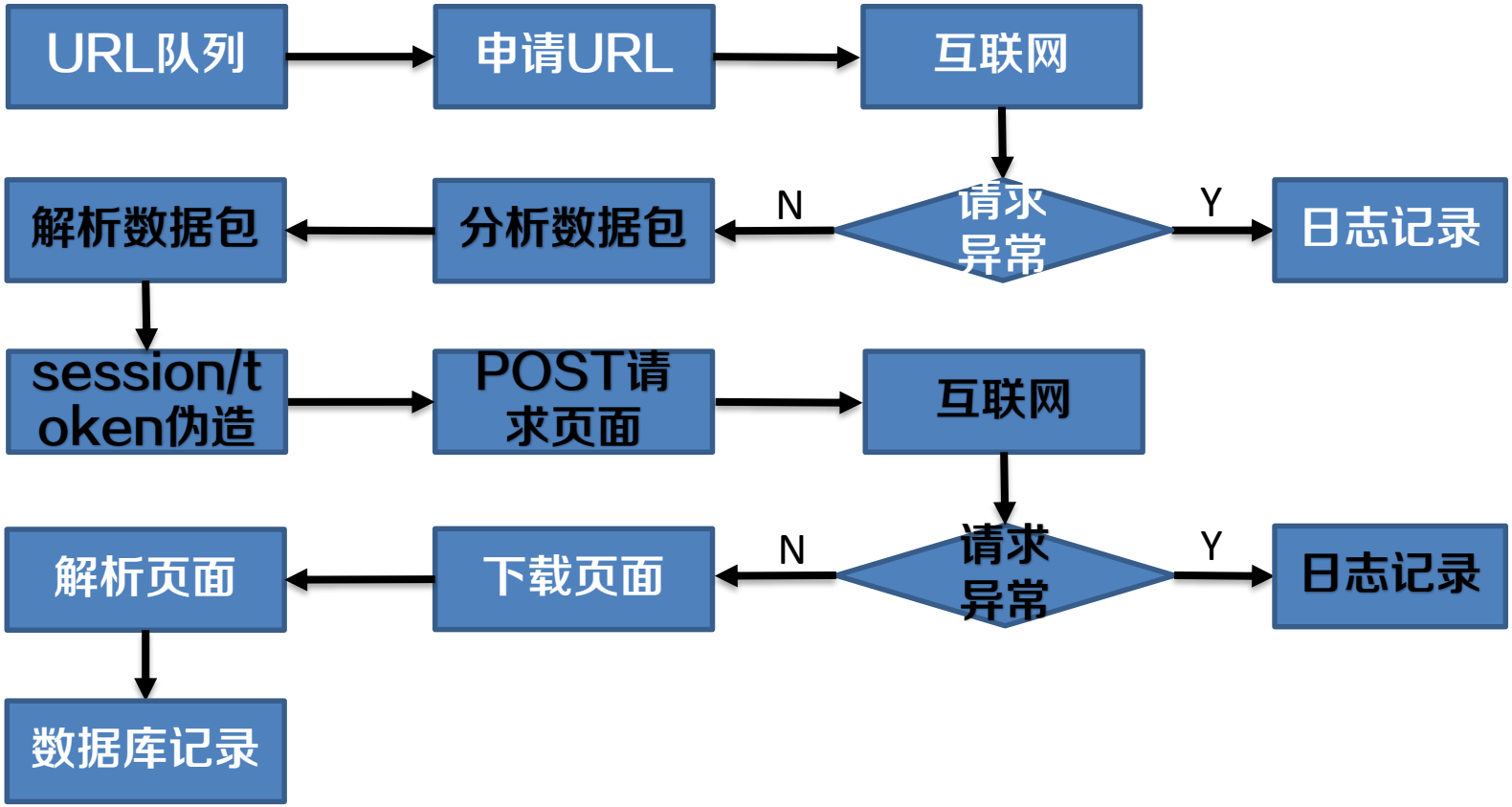
算法原理

- 验证码验证流程图



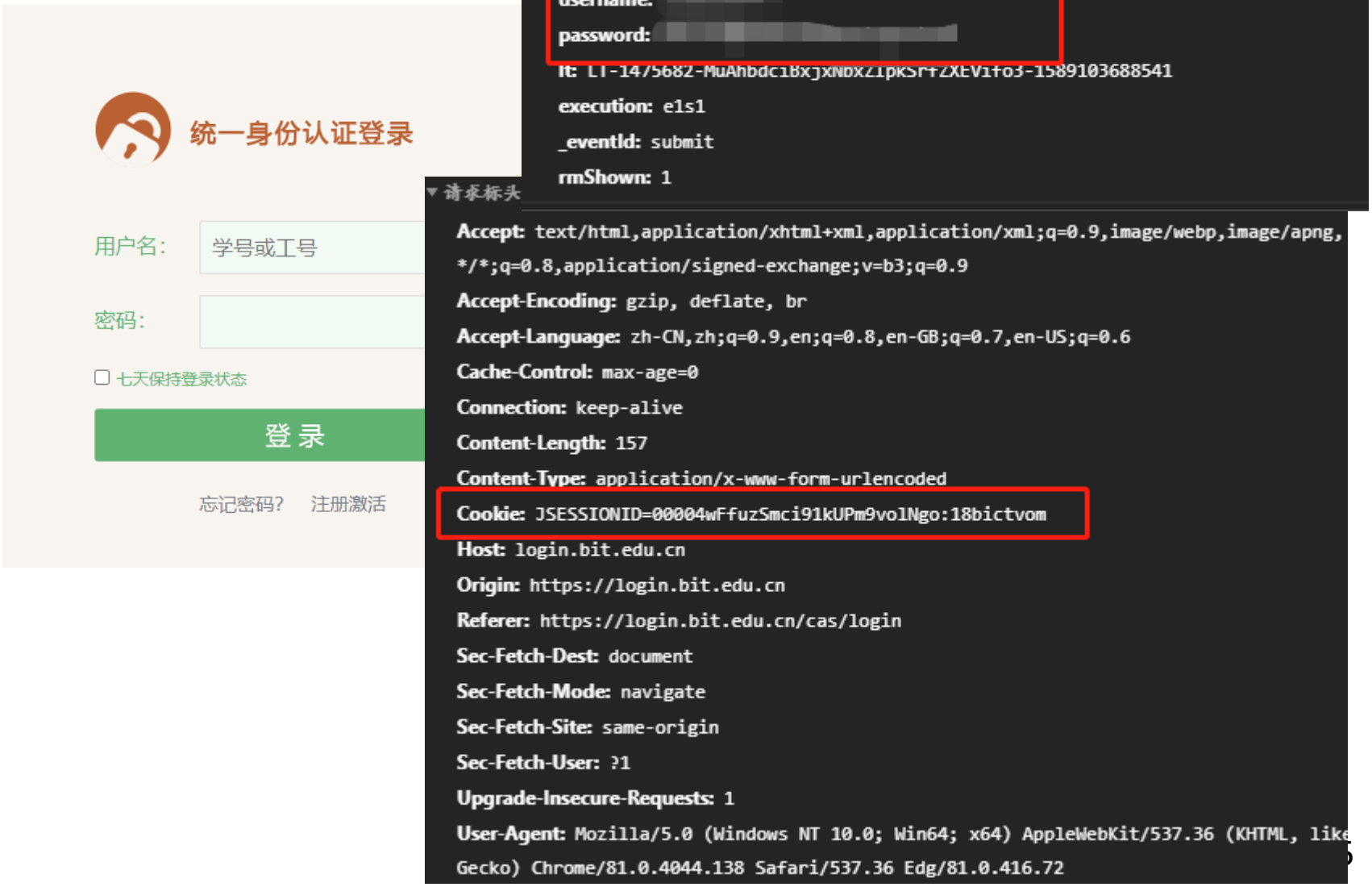
算法原理

- 登录验证流程图



算法原理

- 场景


统一身份认证登录
用户名:
学号或工号
密码:
七天保持登录状态
登 录
忘记密码?
注册激活
表单数据
查看源
编码的视图 URL
username:
password:
lt: LT-1475682-MuAhbdc1BxjxNDxZ1pkSrfZXEVifo3-1589103688541
execution: e1s1
_eventId: submit
rmShown: 1
请求标头
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,
/;q=0.8,application/signed-exchange;v=b3;q=0.9
Accept-Encoding: gzip, deflate, br
Accept-Language: zh-CN,zh;q=0.9,en;q=0.8,en-GB;q=0.7,en-US;q=0.6
Cache-Control: max-age=0
Connection: keep-alive
Content-Length: 157
Content-Type: application/x-www-form-urlencoded
Cookie: JSESSIONID=00004wFfuzSmci91kUPm9volNgo:18bictvom
Host: login.bit.edu.cn
Origin: https://login.bit.edu.cn
Referer: https://login.bit.edu.cn/cas/login
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?1
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/81.0.4044.138 Safari/537.36 Edg/81.0.416.72

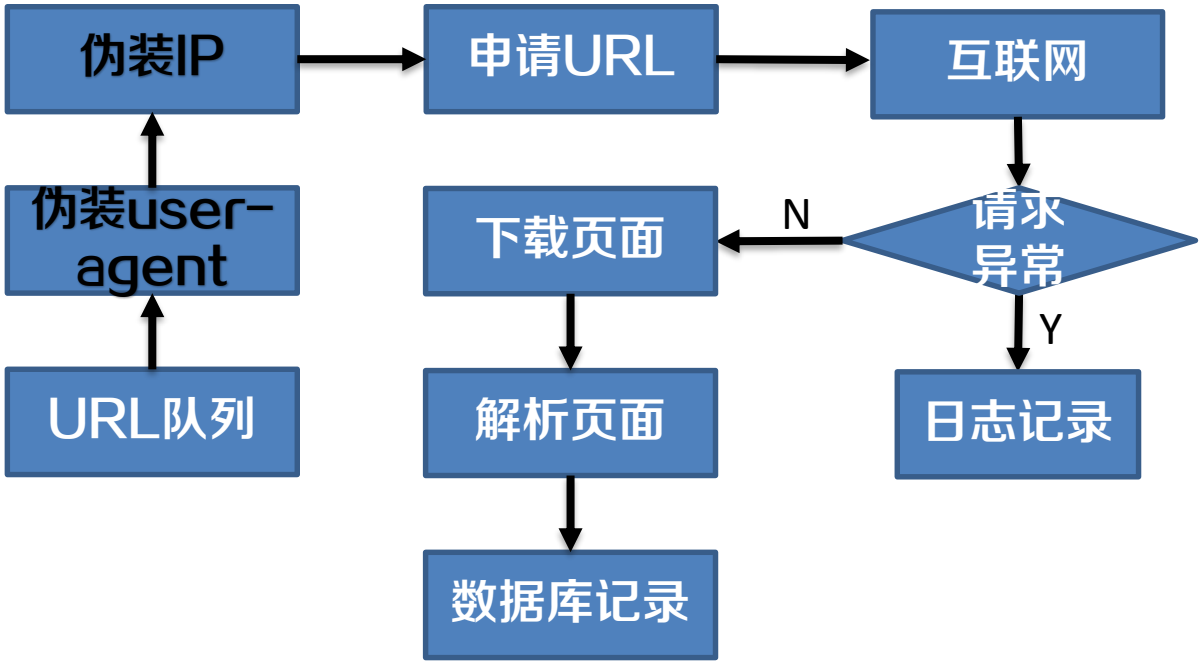
算法原理

- 3. 在http请求header部分user-agent伪装为正常浏览器(chrome浏览器，firefox浏览器)，用户IP不断变换模拟不同地理位置的人浏览网页达到对抗网站设置的同一IP多次访问限制

T	获取设置有同一IP多次访问限制的网页文本信息
I	设置有同一IP多次访问限制的网页url
P	在http请求header部分user-agent伪装为正常浏览器
O	设置有同一IP多次访问限制的网页文本信息

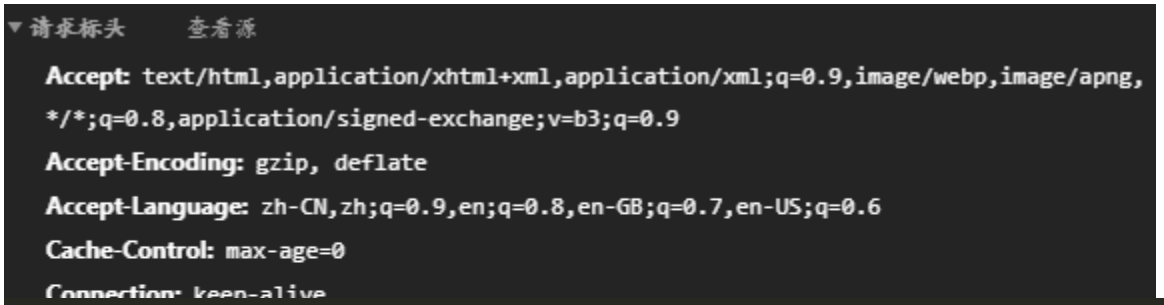
算法原理

- 流程图


伪装IP
申请URL
互联网
伪装user-agent
下载页面
N
请求异常
Y
URL队列
解析页面
日志记录
数据库记录

算法原理

- 场景




```
proxy = random.choice(proxyIP_5.proxy_list)
proxyHandler = urllib.request.ProxyHandler(proxy)
opener = urllib.request.build_opener(proxyHandler, urllib.request.HTTPHandler)
urllib.request.install_opener(opener)
request = urllib.request.Request(link)
ua = random.choice(user_agents_1.user_agents)
request.add_header('User-Agent', ua)
request.add_header('Connection', 'Keep-Alive')
html = urllib.request.urlopen(request, data=None, timeout=500)
return html
```

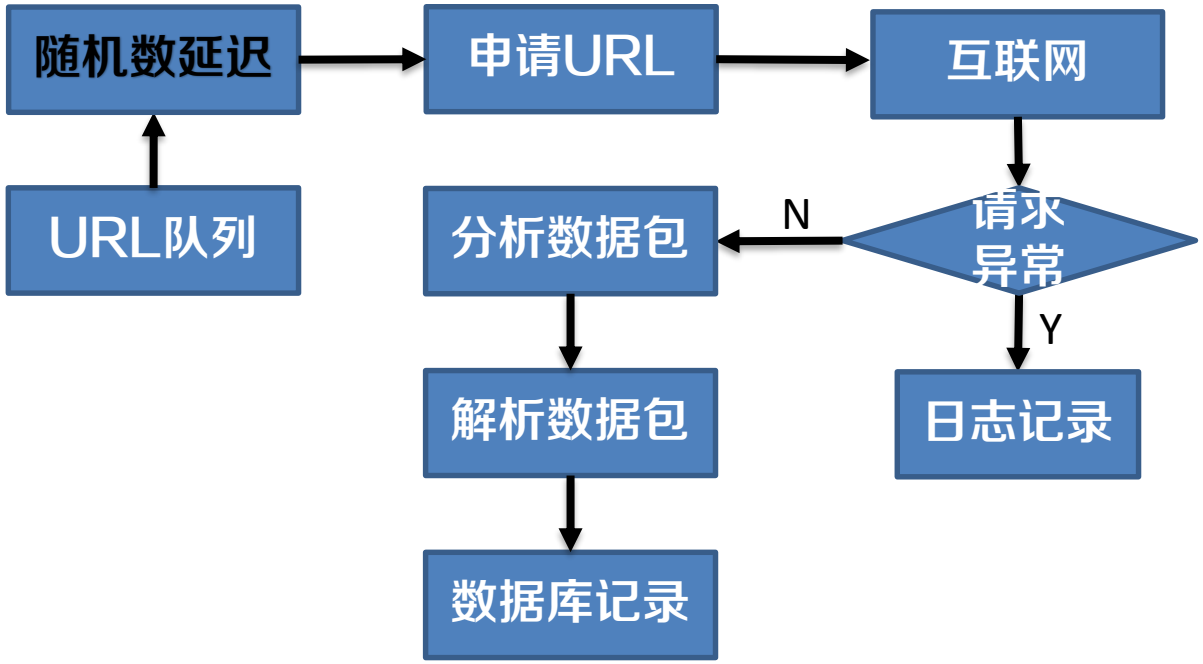
算法原理

- 4. 伪造随机数延迟访问网页模拟人的浏览网站行为达到网站设置的短期内多次访问限制

T	获取设置有短期内多次访问限制的网页文本信息
I	设置有短期内多次访问限制的网页url
P	伪造随机数延迟访问网页
O	设置有短期内多次访问限制的网页文本信息

算法原理

- 流程图



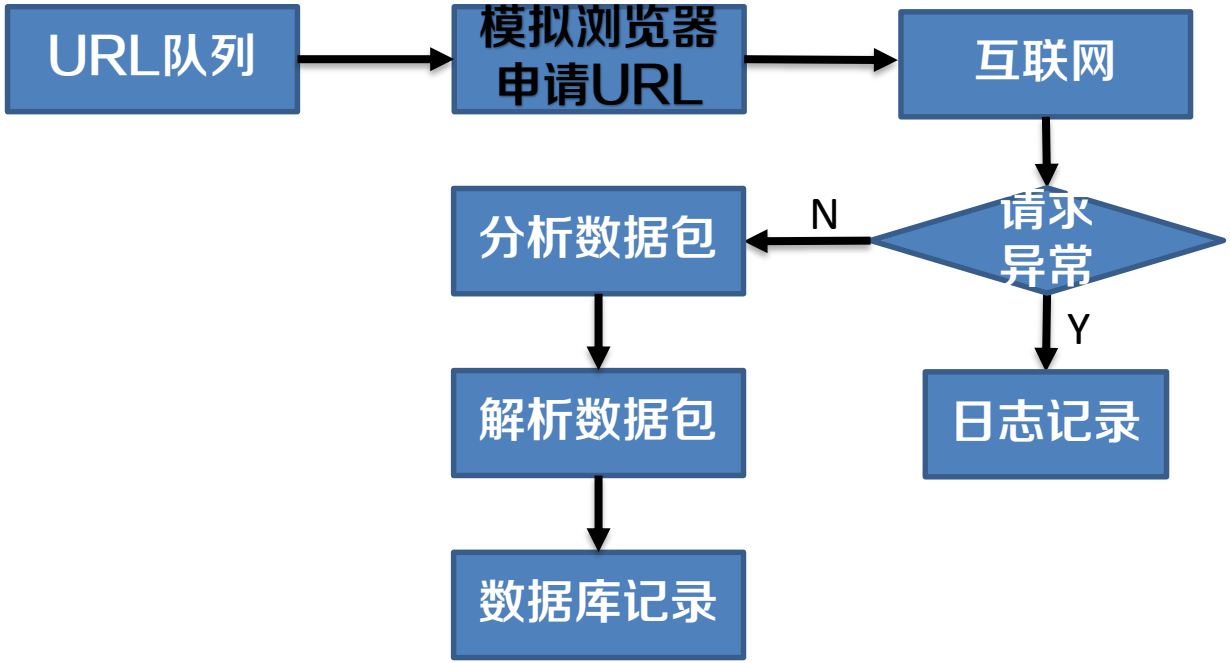
算法原理

- 5. 使用selenium中的driver，模拟不同浏览器访问网页达到网站设置的短期内多次访问限制、网站设置的动态页面加载限制

T	获取设置有短期内多次访问限制、动态加载限制的网页文本信息
I	设置有短期内多次访问限制、动态加载限制的网页url
P	使用selenium中的driver，模拟不同浏览器访问网页
O	设置有短期内多次访问限制、动态加载限制的网页文本信息

算法原理

- 流程图



优劣分析

- 对抗反爬成功率：
- 伪造user-agent<随机数延迟<selenium.driver
- 爬虫速度：
- 伪造user-agent>随机数延迟>selenium.driver

反思总结

- 爬虫初衷是以代码代替人工进行高效率数据获取，反爬手段主要目的在于维护网站服务器，对抗反爬手段必须模拟人的浏览习惯
- 网站访问流量反比于反爬效率
- 爬虫效率反比于对抗反爬手段

参考文献

- https://www.cnblogs.com/yunxintryyoubest/p/10885574.html
- [1]赵茉莉. 网络爬虫系统的研究与实现[D].电子科技大学,2013.
- [2]肖戈林.HTTP协议技术探析[J].江西通信科技,2001,(1):17-24. DOI:10.3969/j.issn.1009-0940.2001.01.005.

谢谢！

大成若缺，其用不弊。大盈若冲，其用不穷。大直若屈。大巧若拙。大辩若讷。静胜躁，寒胜热。清静为天下正。