## 改进

### 反爬改进

由于要使用Scrapy进行测试，不得不大量爬取某个网站的内容。然而，在Scrapy提供的下载中间件机制，可以在请求队列与下载请求之间做一些动作。Scrapy本身也提供了一个ProxyMiddleware，但是它只能使用固定的IP地址。由于固定的IP地址可能遭到IP被禁用的情况，这么一来，你就不能使用本地IP进行相应的爬虫工作了。所以，此时就需要进行反爬虫的实现，以下是使用到的两种基本策略。

1. 伪装user agent

User agent 是HTTP协议的中的一个字段， 其作用是描述发出HTTP请求的终端的一些信息。 服务器通过这个字段就可以知道要访问网站的是什么人了。每个浏览器，每个正规的爬虫都有其固定的user agent，因此只要将这个字段改为这些知名的user agent，就可以成功伪装了。不过，不推荐伪装知名爬虫，因为这些爬虫很可能有固定的IP，如百度爬虫。与此相对的，伪装浏览器的user agent是一个不错的主意，因为浏览器是任何人都可以用的，换名话说，就是没有固定IP。推荐准备若干个浏览器的user agent，然后每次发送请求的时候就从这几个user agents中随机选一个填上去。可以使用以下的一些user agent:

"Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; AcooBrowser; .NET CLR 1.1.4322; .NET CLR 2.0.50727)",

"Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.0; Acoo Browser; SLCC1; .NET CLR 2.0.50727; Media Center PC 5.0; .NET CLR 3.0.04506)",

"Mozilla/4.0 (compatible; MSIE 7.0; AOL 9.5; AOLBuild 4337.35; Windows NT 5.1; .NET CLR 1.1.4322; .NET CLR 2.0.50727)",

"Mozilla/5.0 (Windows; U; MSIE 9.0; Windows NT 9.0; en-US)”

1. 使用免费代理

如果对方用某段时间内某IP的访问次数来判定爬虫，然后将这些爬虫的IP都封掉的话，以上伪装就失效了。对方的这个思路隐含着一个假设：爬虫的访问量必然比正常用户的大很多，因而只要使这个假设不成立就可以了。这时就该代理上场了。所谓代理就是介于用户与网站之间的第三者：用户先将请求发到代理，然后代理再发到服务器，这样看起来就像是代理在访问那个网站了。这时，服务器会将这次访问算到代理头上。同时用多个代理的话，单个IP的访问量就降下去了，于是就有可能逃过一劫。不过，这个方法最大的问题就是找到稳定的代理。在<http://www.xicidaili.com>这个网址上可以找到免费的代理，可能一些不可用，可以使用python代码进行代理抓取以及可用性的测试，将可用的代理保存到相应的文件中。供Scrapy项目进行使用。

### 反爬实现

以上提供的两种基本策略都可以在Scrapy项目中进行使用，主要的改进步骤为：

1. 新建Scrapy项目
2. 重写Scrapy中的middleware的代码，来随机使用伪装的user agent和免费代理
3. 编写免费代理的抓取以及测试的代码，同时保存到相应的txt文件下
4. 在Scrapy项目的设置文件下进行相应的配置设置
5. 针对某个目标网站进行爬取代码的编写