新闻URL及标题爬取Demo

使用说明

本次Demo设计使用简单的Dos操作界面，基本操作如下：

1. 运行文件 `EducationNewsSpider\dist\EducationNewsCollection.exe`
2. 输入需要爬取的页数，程序将会开启4个生产者线程，以及4个消费者线程对`0-输入页面`进行爬取
3. 存入本地文档：`EducationNewsSpider\dist\站点名_page_0-输入页面`，如果对同一站点的同一连续页面进行爬取，结果将会覆盖上次爬取的结果。

存储结果格式如下：

```
{
    "_Announcement__url": "url1",
    "_Announcement__title": "title1"
}
{
    "_Announcement__url": "url2",
    "_Announcement__title": "title2"
}
```

设计思路

本次设计主要基于Request+BeautifulSoup进行开发。

主要可分为以下三个模块：

1. 下载模块
2. 解析模块
3. IO模块

采用多线程异步IO的方式进行设计。

多线程部分采用生产者消费者设计模式。

生产者线程

生产者线程`Crawler`主要负责从任务队列中取任务，并实现下载模块以及解析模块的功能，最后将解析好的结果存入内容池。

下载模块

下载模块由`Crawler`对象中的函数`downLoad_page`完成，主要功能是：

1. 随机产生的请求头
2. 使用`requests`模块请求html页面
3. 下载页面。

解析模块

该模块作为`Crawler`的一个属性存在，在实例化时传入。

该属性需要接收一个生成器。

该模块主要负责：

1. 使用`charset`对请求下来的页面字符集进行推测
2. 使用`BeautifulSoup`模块对请求下来的页面解析
3. 对需要的信息进行格式化
4. 对垃圾页面进行筛查。

总结

生产者线程的主要工作流程如下：

1. 检测任务队列是否为空
2. 从任务队列中取出一条请求任务
3. 调用下载模块请求并下载该页面
4. 调用解析模块解析下载好的页面
5. 将有效页面存入内容池

消费者线程

消费者线程`Parse_man`主要负责将内容池中的对象序列化。主要负责完成IO模块的功能。

IO模块

IO模块主要负责将内容池中的对象序列化。

本Demo的对象序列化操作为：

以json格式将对象存入文本文件中，存入格式如下：

```
{
    "_Announcement__url": 'url',
    "_Announcement__title": 'title'
}
```

总结

消费者线程的主要工作流程如下：

1. 检测内容池是否为空，且生产者进程是否全部结束
2. 如果内容池不为空，从内容池中取出一个对象，并对该对象进行序列化
3. 否则判断生产者是否全部结束
4. 如果生产者全部结束，则消费者线程结束

5. 否则等待

参考文献

1. 周德懋,李舟军. 高性能网络爬虫:研究综述[J]. 计算机科学,2009,36(8):26-29,53. DOI:10.3969/j.issn.1002-137X.2009.08.007.