基于 Python 的专业网络爬虫的设计与实现

姜杉彪，黄凯林，卢昱江，张俊杰，曾志高，刘　强
（湖南工业大学 计算机与通信学院，湖南 株洲 412007）

【摘　要】网络爬虫，又称网页蜘蛛、网络机器人。随着计算机技术的高速发展，互联网中的信息量越来越大，搜索引擎应运而生。传统的搜索引擎会有返回结果不精确等局限性。为了解决传统搜索引擎的局限性，专用型网络爬虫在互联网中越来越常见。同时，专用型网络爬虫具有专用性，可以根据制定的规则和特征，最后只体现和筛选出有用的信息。
【关键词】Python；网络爬虫；数据挖掘；搜索引擎
【中图分类号】TP393　**【文献标识码】**A　**【文章编号】**1674-0688(2016)08-0017-03

0　引言

在搜索引擎的使用过程中，用户认为通用搜索引擎都有一个局限性，那就是在搜索结果中附带太多不必要的信息。用户在使用搜索引擎后，仍然需要人为地从搜索结果中寻找检索最终需要的信息。然而，在互联网飞速发展的状况下，网络信息量突发式的暴增，计算机硬件设备的技术不断进步，网络的信息容量和带宽也是日新月异，在互联网中出现了多媒体和富文本的新技术。随着这些信息的不断增加和积累，通用搜索引擎对类似这种多媒体或者富文本的搜索能力越来越差[1-2]。

为了解决部分用户对信息的检索要求，专用型网络爬虫应运而生，为用户提供特定的信息抓取，开发出不同特性的专用型网络爬虫[3]。本文以网易新闻爬虫实例为引导，对如何开发出专用型网络爬虫及制定不同的爬虫策略进行了深入的研究。

1　爬虫系统需求分析与设计

为了保证网络爬虫系统的开发过程顺利，以及保证最终的开发结果能满足基本的功能需求，必须在开发系统之前进行分析，并设计出符合该系统的代码规范及功能模块等。整个网络爬虫系统均使用模块化设计，一个功能类作为一个功能模块。这样做的目的是一方面可以便于代码的维护，另一方面可以增加代码的重用性。通过将整个系统进行模块划分，每个功能模块只实现一个功能，最后所有的模块功能完成后，整个网络爬虫系统就能实现当初进行定义的系统功能[4-5]。本系统的需求分析的任务是通过调查特定用户的上网行为习惯，开发符合一类上网用户使用的专用型网络爬虫，根据用户的功能需求明确系统需要实现的各个功能。并且，在设计系统的同时，需要考虑系统今后的维护及改进问题。本文以网易新闻爬虫系统为例，探讨专用爬虫系统的设计与实现。

1.1　功能性需求分析

网易新闻爬虫的具体功能包括对新闻标题、新闻 ID、新闻来源等信息进行抓取并存入数据库中。网易新闻爬虫需要抓取的 URL 链接是变化的，而不是固定的，因此在爬虫的 URL 策略中，需要解决 URL 链接的访问策略和去重。原站点网易新闻中的各类新闻，根据分析可以看出，所有新闻一旦发布就不会对新闻内容进行二次更新，因此网易新闻爬虫最终的抓取结果不需要对数据库

【基金项目】2015 年湖南省科技计划项目(项目编号：2015GK3024）“基于物联网的药品质量安全追溯系统”；湖南工业大学教学改革项目(项目编号：2013B11）“基于移动互联网的网络教学资源建设与成效研究”；湖南省教育厅科学研究项目（编号：13C036）“WEB 数据挖掘在网络学习资源推荐系统中的应用研究”。
【作者简介】姜杉彪，云南楚雄人，湖南工业大学计算机科学与技术专业在读本科生，湖南省高校大学生百佳党员，研究方向：网络数据分析与处理；黄凯林，广东惠州人，湖南工业大学计算机科学与技术专业 2011 级本科生，研究方向：网络技术应用；曾志高，博士，湖南工业大学副教授，硕士生导师，研究方向：模式识别，数据处理。

中已存在的新闻进行更新操作，只需将网站更新的新闻进行入库操作。网易新闻爬虫架构图如图 1 所示。

当中，Scarpy Engine 是一个抽象的爬虫框架引擎，控制爬虫的所有操作；Spider 类为爬虫主要的页面处理模块类，Item Pipleline 组件可以实现清理 HTML 数据，或者验证抓取的数据。

1.2 爬虫功能设计

网易新闻爬虫功能的具体设计思路如下：首先，网易新闻爬虫不会与前端页面进行直接交互，而是通过系统设置的定时任务，对爬虫进行定时执行，从而达到一个自动定时向原站点抓取新闻的爬虫功能。其次，网易爬虫被系统定时任务激活后，爬虫根据定义的 URL 规则对原站点的站点目录进行正则匹配，符合正则匹配的 URL 链接则进行抓取，并且对抓取结果进行过滤和提取需要的信息。最后，将抓取的新闻信息和数据库进行对比，数据库中没有该新闻则进行插入更新，有则停止爬虫，其流程如图 2 所示。

2 爬虫页面抓取模块

爬虫的页面抓取模块是爬虫程序的第一个执行模块。在页面抓取实施之前，需要获得目标站点的状态，以及 DNS 解析和记录去重等各种功能。在爬虫进行页面抓取的时候，必须保证目标站点的状态是可抓取的。因此，部分目标站点如果要用户登录后才可以请求相应的服务器资源，则必须对该目标站点进行模拟登录后才可进行页面抓取。

通过模拟登录可以解决目标站点的登录限制。模拟登录采用完全遵循目标站点的登录规则，使用用户的用户名、密码、cookies 和伪造 User-Agent 及 Referer。最后通过返回的 session 与服务器进行请求交互并且进行页面抓取，完成整个页面抓取的过程。

DNS 解析和 URL 记录去重是页面抓取模块比较重要的一环。当大量的页面需要进行页面抓取，以为页面抓取都是通过 URL 地址进行抓取，因此在请求 URL 时，需要对 URL 进行解析。当需要解析 URL 的记录非常多时，DNS 解析就有可能是页面抓取的瓶颈部分，要处理 DNS 解析的瓶颈，最直接的方法是对 DNS 解析结果进行本地缓存。

记录去重是一种对已经抓取的 URL 地址进行记录去重。页面抓取在一定的时间内，只需要进行一次抓取。

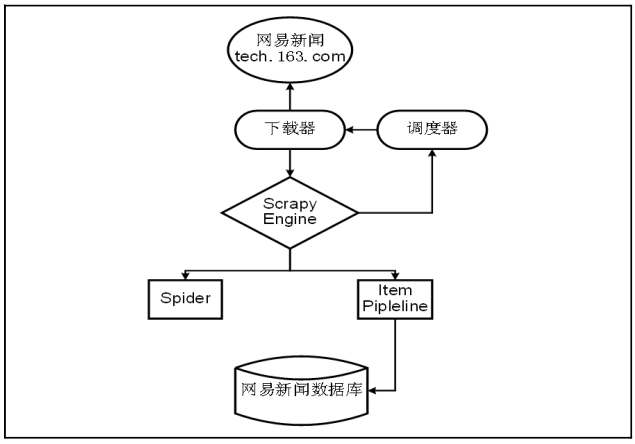

图 1 网易新闻爬虫架构图

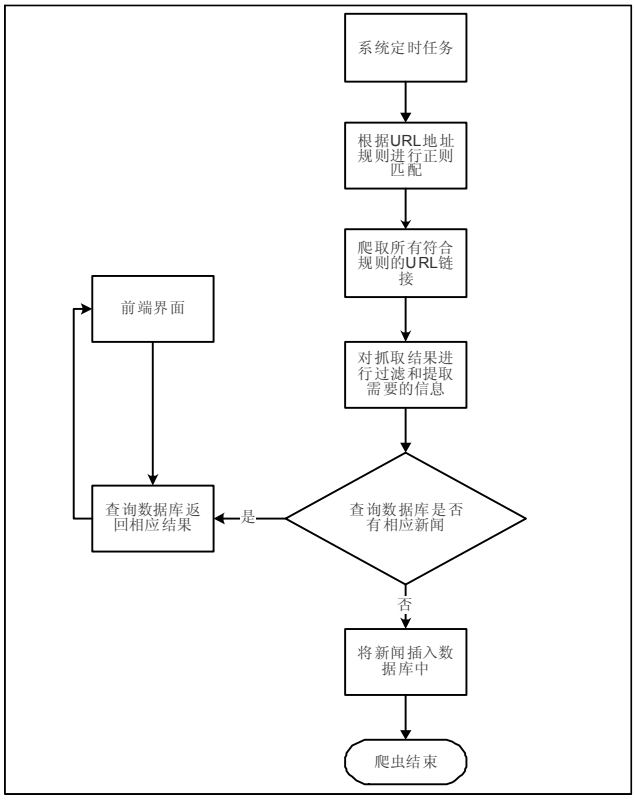

图 2 网易新闻爬虫流程图

为了避免重复抓取的现象造成系统性能差及提升信息的高效性，因此在页面抓取过程中，做好对应的记录去重。

3 爬虫页面处理模块

对页面抓取结果需要进行相应的处理。页面处理包含了对 HTML 源码的处理和过滤，过滤出需要的信息，最后对过滤出的信息进行整合，并进行入库操作。通常利用正则表达式来进行页面处理，但是当 HTML 源码非常多时，在编写正则表达式时会显得非常困难。

接下来利用 XPath 对 HTML 源码进行过滤操作，通常不同的处理需求，定义不同的 XPath 语法。例如，get_title 方法的 XPath 语法如下所示：title=responsexpath（"/ html/head/title/text ()").extract()。仅需通过该 XPath 语法，就能过滤出新闻的标题，而不需要编写复杂的正则表达式对象。再如，get_source 方法的 XPath 语法如下所示：source =responsexpath ("//div [@class='ep-time-sourec-DGray']/text()").extract()。经过 get_title 和 get_source 这几种方法后，页面处理模块最终可以得到一条新闻中例如新闻编号、新闻标题、新闻来源等这些信息。得到这些原始数据后，将这些数据整合为一个列表，传递到爬虫入库模块中。流程如图 3 所示。

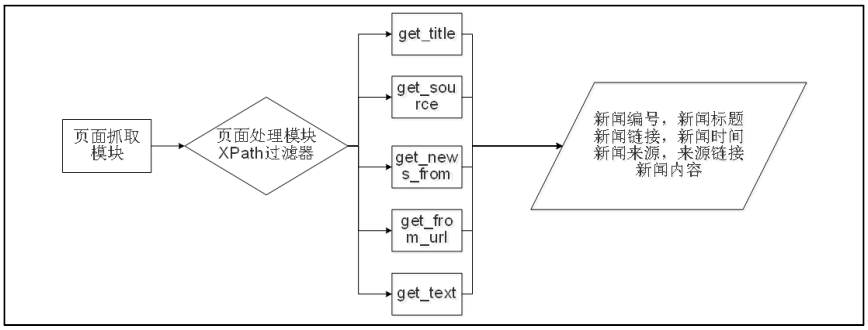

图 3　网易新闻爬虫页面处理逻辑图

图 4　网易新闻界面

4　爬虫系统功能实现

网易新闻的功能界面中，由于网易新闻的不定期更新，为了节约服务器的资源利用，设置网易爬虫的工作时间是每一个小时自动执行一次。在网易新闻的功能界面中，显示的是当天所有的新闻，点击新闻的标题，就可以跳转到原网站网易新闻的对应新闻界面中（如图 4 所示）。

爬虫程序根据定义的 URL 规则，可以对原站点所有符合正则匹配的 URL 链接进行抓取处理。

```
day = time.strftime('%m%d')
        name = "news"
        allowed_domains = ["tech.163.com"]
        start_urls = ['http://tech.163.com']
        rules = [
            Rule (LinkExtractor (allow=r"/15/"
+ day + "/\d+/*"),
    callback="parse_news", follow=True)]
```

5　结论

本文通过使用 Python 语言的库的调用，实现了一个简单的网络爬虫系统。页面抓取的效率及结果也是对爬虫性能的一个考验。整个互联网的站点数量庞大，所有站点的开发过程中遵循的原则不一致，代码风格不一致。因此对整个爬虫的页面抓取模块是非常大的挑战，需要对不同的代码风格做不同的处理，但是最终得到的结果需要是一样的。在将来的工作中，我们将进一步提高爬虫的速度。

参　考　文　献

[1] 李勇，韩亮．主题搜索引擎中网络爬虫的搜索策略研究[J]．计算机与数字工程，2008，228(10)：50-53.

[2] 罗刚，王振东．自己动手写网络爬虫[M]．北京：清华大学出版社，2010.

[3] (美)Miguel Grinberg．Flask Web 开发[M]．安道，译．北京：人民邮电出版社，2015.

[4] Magnus Lie Hetland．Python 基础教程(第二版)[M]．司维，曾军崴，谭颖华，等，译．北京：人民邮电出版社，2010.

[5] 叶允明，于水，马范援，等．分布式 Web Crawler 的研究：结构、算法和策略[J]．电子学报，2002，30(12)：2008-2011.

[责任编辑：钟声贤]