## Java第二阶段报告

PkuJava08组 组员：许芳芳，李欣，郭转转

## 1.项目题目

基于爬虫技术的社交网络平台的好友推荐系统

## 2.项目简介

利用爬虫技术给豆瓣社交网络的用户推荐好友。根据豆瓣中每个用户喜欢的书，电影，音乐，关注的小站，常去的小组，豆列，关注的人，和被关注的人这些作为推荐的依据，向用户推荐他们有可能及其感兴趣的人或者认识的人。

这里我们以北京大学小组这个组作为数据源，通过爬取相关信息，了解其小组中用户之间的关系，向用户推荐好友。

## 3.项目架构

* 爬取技术：selenium-java-2.40.0.jar

Selenium是一个模拟浏览器，进行自动化测试的工具，它提供一组API可以与真实的浏览器内核交互。Selenium是跨语言的，有Java、C#、python等版本，并且支持多种浏览器，chrome、firefox以及IE都支持。Selenium是一个开源的自动化测试框架，Selenium也可以作为爬虫工具。Selenium可以模拟事件，如点击、滚动等，实现数据自动爬取。

* 数据库：MySQL Hibernate DAO

本实验的数据库使用MySQL，总共包含10个表：

1. Basicinfo：用户基本信息表，存储用户名、ID、地址等基本属性，并使用tag标记该用户的属性是否已经完成爬取，如该用户的好友名单已经爬取完成，则将对应的tag记作1，否则记作0。
2. Bookinfo：用户ID、该用户读过的书的标签及数量。不记录书名的原因是，书的数量太多，用户之间的重叠度较低，可能对好友推荐意义不大，因此使用的是书的类别信息。
3. Doulist：用户ID、该用户关注的豆列ID及名称和豆列大小。
4. Focusinfo：关注者ID和被关注者ID。
5. Groupinfo：用户ID、所在小组ID及名称和大小。
6. Likeinfo：用户ID、喜欢的条目的ID和名称。
7. Minsiteinfo：用户ID、小站ID和名称。
8. Movieinfo：用户ID、电影类别和数量。
9. Musicinfo：用户ID、音乐类型和数量。
10. Parserconfig：目标爬取的页面数量、当前以及完成爬取的页面数量。

DAO(Data Access Object)对象是模块化的数据库访问组件，DAO对象通常包括：对持久化类的基本CRUD操作（插入、查询、更新、删除）操作。Spring对Hibernate的DAO实现提供了良好的支持。建立好数据库后，可以使用MyEclipse通过hibernate自动生成DAO层，对数据库增删改查的SQL操作进行封装。

* 抓取类 Parser

对需要抓取的每种信息建立一个爬取类，通用的功能写在一个共同的抽象父类AbstractParser中，这个父类实现了一个接口ParserImp，其中有一个空的负责数据抓取的方法parser，所以每个爬取类需要实现这个parser方法。对于电影、书籍、音乐这三类信息的抓取方式非常相似，有很多共同的操作和方法，因此这三个抓取类继承一个父类InterestParser（InterestParser继承了AbstractParser）。

* 控制类 ParserControl

负责开启抓取线程。

* 工具类 Utils

LoginHelper：该类用于模拟用户登录。

MonitorHelper：该类用来监视进程队列中有无停止不动的线程，如果有，就重启该线程。

ParserThread：用于检查爬虫是否卡住，如果卡住则重启爬虫。

ReflectHelper：用于在爬虫卡住时获取爬虫类名，以便重启该类。

## 4.Demo

我们已经基本完成了初始版本，下面以抓取用户基本信息为Demo进行介绍。这一部分完成的任务是进入“北京大学”小组的成员列表网页，抓取该页面所有成员的基本信息，包括用户昵称、用户ID、地址等。

1. 进入“北京大学”小组成员列表网页

通过"http://www.douban.com/group/pku/members" url进入。

1. 网页自动翻页

由于该网页每页最大的用户数量为35个，观察其url的变化“http://www.douban.com/group/pku/members?start=35”，发现要实现网页翻页可以直接修改url中“start=”后的参数即可，每次加35。

1. 抓取用户列表

通过观察用户列表的url，可以调用findElements方法获取列表路径

driver.findElements(By.*xpath*("//div[@class='member-list']//ul//li"))

1. 抓取用户昵称，用户地址

同理，可以通过

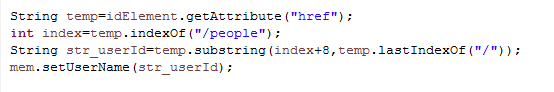
ele.findElement(By.*xpath*("./div[@class='name']//a"));获取用户昵称；

通过

driver.getElement(By.*xpath*("./div[@class='name']//span[@class='pl']"), ele);获取用户地址

1. 抓取用户ID:

通过观察可知道ID号一般在url中“/people/”和“/”之间，所以我们通过截取中间的字符串存入用户ID一项。代码如下：



## 5.技术难点

1. 断点爬取：

例如已经爬取了100位用户的信息，当你重新启动程序时，它会在从上次爬取结束的地方继续爬取下一位，即101位用户；

1. 避免多次输入验证码：

当爬取数据时，如果爬取的过快，会需要输入验证码，则可以将爬去时间放慢，避免出现这样的情况；

1. 防止浏览器卡死：

在很多情况下，浏览器在爬取数据时，会在某个url停留较长时间，这样会影响爬取数据的效率，所以我们可以对后台进行扫描，如果url在6s之内没有跳转，我们就默认浏览器属于卡死状态，然后跳过此次数据爬取，转向下一个，进行数据的爬取。

1. 利用多线程爬取多种信息：

为了提高爬取数据的效率，可以采用多线程爬取多种数据，例如可以同时开启三个线程即浏览器，分别爬取，用户基本信息，关注的人信息以及被关注人的信息。

1. 速度优化：

在爬取个人信息时，除了ID还会有PIC图片这一数据元素，但是这一项对于我们并不重要，如果爬取还会影响爬取速度，则为了优化速度，我们将不下载图片。

1. 使用cookies：

在豆瓣中查看一些用户信息，是需要进行用户登录才可以看到，并且每次运行程序都需要进行输入，十分繁琐，则这时候需要将登录用户名和密码保存到cookies中，就可以解决这个问题。

## 6.任务分工

* 许芳芳：

搜集相关资料；用户的基础信息的爬取；技术难点攻克：断点爬取，避免多次输入验证码；数据处理分析。

* 李欣：

搜集相关资料；用户兴趣（书，音乐，电影）信息的爬取；技术难点攻克：防止浏览器卡死，利用多线程爬取多种信息；数据处理分析。

* 郭转转：

搜集相关资料；用户关注的小站等信息的爬取；技术难点攻克：速度优化，使用cookies；数据处理分析。

## 7．时间点

* 2015年12月3号－2015年12月9号：对于爬虫技术进行调研，查找相关文献以及论坛等，对任务进行划分；
* 2015年12月10号－2015年12月16号：对于自己所分配的任务练习写较为简单的demo，并提出技术难点；
* 2015年12月17号－2015年12月23号：各自任务的完善，以及提出解决难点的方案，并对推荐算法进行调研；
* 2015年12月24号－2015年12月31号：基于推荐算法的数据处理，得出结果，分析优劣；以及完成PPT。