基金项目论文

基于 Python 的文献检索系统设计与实现

杜 兰[1]，刘 智[2]，陈琳琳[1]

（1. 南京理工大学紫金学院，江苏 南京 210023；2. 南京电子技术研究所，江苏 南京 210039）

摘 要：毕业设计是大学本科教育的一个重要教学活动，既能检验本科阶段学习成果，又能提升实践创新能力。而毕业设计需要学生掌握所毕业课题的学术动态，这要求学生能正确有效地进行文献检索，获取最新发表的文献资料。现如今，大多数学生采用的是手工操作的方式。而海量数据带来的“信息过载”问题，增长了用户检索时间，降低了查准率，严重影响效率。因此，为了帮助学生在浩瀚的文献库里找到满足自己专业化、个性化需求的资料，本文系统首先利用 Python 爬虫获取文献，实现自动化文献检索和下载。然后基于协同过滤推荐算法，实现基于检索的个性化推荐。该系统能为学生提供准确高效的文献检索服务，提升学生毕业设计质量，是一项值得推广的技术。

关键词：文献检索；爬虫；毕业设计；推荐

中图分类号：TP391. 41 **文献标识码：**A **DOI：**10.3969/j.issn.1003-6970.2020.01.012

本文著录格式：杜兰，刘智，陈琳琳. 基于 Python 的文献检索系统设计与实现[J]. 软件，2020，41（01）：55–59

Design and Implementation of Thesis Retrieval System Based on Python

DU Lan[1], LIU Zhi[2], CHEN Lin-lin[1]

(1. *Nanjing University of Science and Technology ZiJin College, Nanjing* 210023, *China*;
2. *Nanjing Research Institute of Electronics Technology, Nanjing* 210039, *China*)

【Abstract】: Graduation design is an important teaching activity in undergraduate education. It can not only test the results of undergraduate study, but also improve the ability of practice and innovation. Graduation design requires students to master the academic dynamics of their graduation projects, which requires students to correctly and effectively retrieve documents and obtain the latest published documents. Nowadays, most students use manual operation. However, the problem of "information overload" caused by massive data increases the retrieval time of users, reduces the accuracy rate, and seriously affects the efficiency. Therefore, in order to help students find information to meet their professional and personalized needs in the vast literature library. Firstly, the system uses Python crawler to obtain documents and realize automatic document retrieval and download. Then, based on collaborative filtering recommendation algorithm, personalized recommendation based on retrieval is realized. The system can provide accurate and efficient literature retrieval service for students and improve the quality of graduation design. It is a technology worth promoting.

【Key words】: Thesis retrieval; Crawler; Graduation design; Recommendation

0 引言

众所周知，文献检索是大学生毕业设计过程中不可或缺的研究手段。它能让学生在前人的成果上找到起点，激发潜能，拓宽思路，培养创新能力。现如今，随着图书馆数字化进程的发展，现在学生普遍采用的是通过联机检索方式来检索并下载文献。当前检索方式存在如下问题：第一，目前主要的检索关键字是题名、责任者、关键词等，检索到文献后，还需要再次点击链接进入到详细页面才能下载全文，费时费力。第二，在大数据时代，学术资源急速增长，例如知网（CNKI）上已经高达亿万条记录。而学生本身对相关领域的词汇量储备少，这会导致学生在检索时缺乏检索词，加大检索难度。第三，文献领域多样性会导致用户搜到大量无用信息，浪费大量时间，到最后还是搜不到需要的文献。

基金项目：江苏省高校自然科学研究项目(批准号：19KJB520039)；江苏省高校哲学社会科学研究项目(批准号：2019SJA2056)

作者简介：杜兰(1986–)，女，副教授，主要研究方向：人工智能、区块链；刘智(1986–)，女，工程师，主要研究方向：人工智能；陈琳琳(1981–)，女，讲师，主要研究方向：模式识别与人工智能。

因此迫切需要实现一个高效精准的文献检索系统，一方面能实现自动化的高效检索，减少手工操作；另一方面能提供个性化的精准推荐，主动为用户帮助学生快速高效地从文献浩瀚的海洋里找到自己需要的资料。

目前，已有部分学者展开相关研究，但杨洋等人的研究侧重如何避免电子学术资源的重复下载，未见自动化下载和推荐相关的研究[1]。刘爱琴等是针对学术资源网站数据结构化的特征，利用用户输入的爬虫的关键字，进行一次检索数据里本体属性的动态分析，形成关联关键词网络，将文献资料本体的所有相关数据来源 URL 链接同时返回到一次检索结果中。尽管该研究用关联关键词网络将检索界面与更多的文献 URL 地址反馈到同一页面，提升了检索效率。但是没有考虑用户历史偏好数据，不能满足用户的个性化需求[2]。因此，本文基于用户的历史行为数据，使用最经典应用最广泛的协同过滤算法[3]从海量数据中分析出用户感兴趣的文献，推送给用户个性化的推荐列表。本文做出如下工作：

（1）设计一个文献爬虫程序[4]，检索并下载文献。

（2）应用协同过滤算法设计文献推荐方案，为用户提供个性化推荐。

1　系统设计

本系统基于 Django 框架标准采用三层架构模式，表现层-模板(Template)、业务逻辑层-视图（Views）和数据存取层-模型（Model），其架构如图 2 所示。具体来说，表现层的功能主要是展示使用 html5 前端开发语言编写的用户的个人信息、文献信息、推荐信息等页面。业务逻辑层主要有两个作用，一是存放数据存取模型，二是调取正确的模板的相关逻辑功能，比如网络爬虫、推荐引擎、文献信息管理、用户信息管理等。其中网络爬虫的主要功能是根据用户检索条件去实现自动化检索和下载。推荐引擎的主要功能是根据训练好的推荐算法模型计算出每个用户喜欢的文献排名，并存储计算结果等待推荐系统调用。数据存取层主要存储、读取和修改用户信息、文献信息、用户评分信息等各种数据。

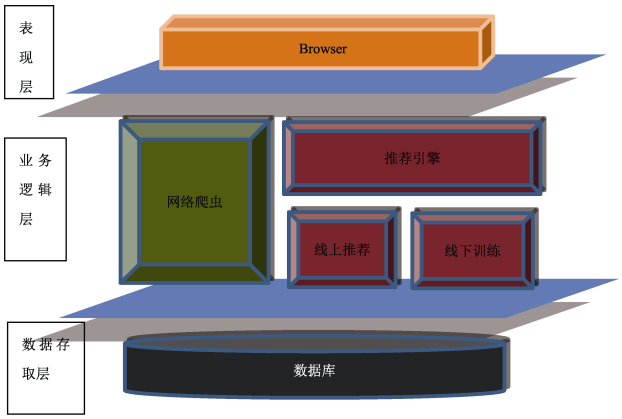

图 1　系统架构图

Fig.1　System architecture diagram

文献检索系统的 3 个过程如下：

（1）创建索引与检索索引：创建索引与检索索引是一切检索的开始，不可或缺，在有了索引之后，才能根据索引建立请求。创建索引,从所搜索的资源网站的请求条件中提取有用信息并重新显示成用户易懂的格式，然后创建索引，大致分为作者，主题，引用文件等几个部分。检索索引，根据用户传输的参数形成查询语句，结合索引转化为网络请求需要的格式，然后向网站发送请求。

（2）网络爬虫：根据用户输入的条件开始检索，在抓取的过程中，首先要发送请求，判断的资源网站的请求参数，遵循资源网站的请求顺序和请求规则。在请求的过程中要注意不影响网站的正常运行，遵循资源网站的爬虫规则，否则会被禁止。在爬取页面的过程中要注意采取 IP 代理，适当延长访问间隔。获得数据的时候，还应该进行筛选与分类，取出自己需要的字段，排除掉不需要的字段。

（3）推荐引擎：根据用户的文献搜索历史、下载记录、评价反馈等数据，使用协同过滤算法，为用户生成个性化推荐列表。

2　关键功能实现

2.1　爬虫实现

（1）创建索引

按（a）主题、（b）关键词、（c）篇、（d）摘要、（e）全文、（f）被引文献、（g）中图分类号、（u）单位、（z）作者分出七个维度检索索引。在用户进行检索的时候，会给出该界面提示，每个提示都有一个代表字母，用户直接输入条件前方的字母就可以，省略去写出具体索引的过程，极大方便了用户的操作，而且该条件还可以填写多个，用户通过输入代表字母，之间通过空格分隔开，以便区分出是几个检索条件。

（2）使用 request 抓取网页

网页抓取模块是通过 Python 包封装的 requests 方法对网页发送请求的模拟访问，然后直接抓取服

务器数据。主要实现过程是把用户输入的条件进行整理，结合索引，将条件转换为请求对应的参数，通过 requests 方法向服务器发送一段请求，在请求正确发送之后，服务器端返回结果。

为了以防请求超时在 post 方法中追加 timeout 字段，自定义一个请求的最大访问等待时间。具体代码如下：

```
static_post_data = {
'action': '',
'NaviCode': '*',
'ua': '1.21',
'isinEn': '1',
'PageName': 'ASP.brief_default_result_aspx',
'DbPrefix': 'SCDB',
'DbCatalog': '中国学术期刊网络出版总库',
'ConfigFile': 'CJFQ.xml',
'db_opt':
'CJFQ,CDFD,CMFD,CPFD,IPFD,CCND,CCJD',    #
搜索类别（CNKI 右侧的）
'db_value': '中国学术期刊网络出版总库',
'year_type': 'echar',
'his': '0',
'db_cjfqview': '中国学术期刊网络出版总库,
WWJD',
'db_cflqview': '中国学术期刊网络出版总库',
'__':    time.asctime(time.localtime())    +    '
GMT+0800 (中国标准时间)'
}
```

（3）使用 beautifulsoup4 库分析提取数据

beautifulsoup4 库对文字字段的获取特别方便，在定位获得字段内容之后，从当前页面获得的是通过条件检索出来的符合用户条件规范的文献。

以知网为实验平台，在文献检索发现返回的只是论文的概要信息，而访问详情页需要发送三次请求。第一次请求获取到服务器验证响应的参数信息。第二次请求根据第一次获取到的参数信息，再此请求服务器，在服务器端验证。在两次访问之后，访问详情页才会获得信息不被拒绝，前两次请求在服务器验证注册之后就可以访问详情页面获得信息了。由此整理出详情页的获取方式，部分源代码如下：

```
self.session.get(
    'http://i.shufang.cnki.net/KRS/KRSWriteHandler.
ashx',
    headers=HEADER,
    params=params)
self.session.get(
    'http://kns.cnki.net/KRS/KRSWriteHandler.
ashx',
    headers=HEADER,
    params=params)
page_url = 'http://kns.cnki.net' + page_url
get_res=self.session.get(page_url,headers=HEAD
ER)
```

（4）处理数据

爬虫爬取下来的数据可以存入 Excel 表中，一共存储三个文件，分别是所有文章的下载链接、文章的简要介绍和文章的详情信息。文章详情信息按照序号、题名、作者、单位、关键字、摘要 、来源、发表时间等字段进行存储，在存储时选择单线程，因为多线程使用 Excel 会导致数据的丢失和错乱。所以在存储或者查看的时候不能对表格信息进行改动，以防出现错误，因此需要获得的数据先存入一个字典之中，通过 sheet.write 方法将文献信息写进去，部分源代码如下：

```
for i in range(0,3):
    self.reference_list.append(self.single_refence_
list[i])
    self.reference_list.append(self.orgn)
    self.reference_list.append(self.keywords)
    self.reference_list.append(self.abstract)
    for i in range(3,6):
    self.reference_list.append(self.single_refence_
list[i])
    if config.crawl_isDownLoadLink=='1':
    self.reference_list.append(self.download_url)
    xuhao=self.single_refence_list[0]
    title=self.single_refence_list[1]
    author=self.single_refence_list[2]
    laiyuan=self.single_refence_list[3]
    fabiaoDate=self.single_refence_list[4]
    shujuku=self.single_refence_list[5]
```

在导出表格之后，还要将爬取到的信息存储到数据库之中，通过导入 pymysql 包实现对 MySQL 数据库的操作，具体的代码如下所示：

```
    sql1 = "insert into detail values (NULL ,%s,
%s,%s,%s,%s,%s,%s,%s,%s,%s,%s)"
    try:
        self.cur.execute(sql1, [xuhao, title, au-
thor, self.orgn, self.keywords
            , self.abstract, laiyuan, fabiao-
Date,shujuku,self.download_url,downnum])
        self.con.commit()
    except:
        pass
```

2.2　推荐模块

本文采用基于用户的协同过滤算法，它是根据用户历史的兴趣偏好，首先找到与目标用户最相似的用户（最近邻居），然后把邻居用户喜欢的而目标用户没有接触过的物品推荐给目标用户。它主要有

三步：（1）用户评价；（2）搜寻近邻；（3）生成推荐列表[5]。

（1）用户评价

用户在文献检索系统中评价打分，形成用户历史数据。

（2）用余弦相似度寻找近邻

使用余弦相似度来计算兴趣相似度，如公式 1 所示，其中 A 和 B 分别代表两个目标物品的坐标，该公式的最终式为余弦公式在 N 维向量中的公式推导结果。

$$similarity = \cos(\theta) = \frac{A \cdot B}{AB} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2 \times \sum_{i=1}^{n} (B_i)^2}} \quad (1)$$

代码实现如下：

```
def cosine(n_x, yr, min_support):
#初始化计算用矩阵,其中 freq 用以统计目标出现交集次数，prods 为交集物品乘积累加矩阵，sqi 与 sqj 分别为为两目标各自物品平方累加矩阵。如果有相同评价物品少于 min_sprt 次，则直接相似度置为 0。
        denum = np.sqrt(sqi[xi, xj] * sqj[xi, xj])
        sim[xi, xj] = prods[xi, xj] / denum
        sim[xj, xi] = sim[xi, xj]
        #返回相似度矩阵
        return sim
```

（3）生成推荐列表

把相似度前 N 名的用户喜欢的物品的评分加权平均，然后按评分从高到低排序给目标用户生成物品推荐列表。

3　运行结果

（1）爬虫检索与下载文献

以知网为实验平台。在选择好主要索引信息之后，会进行次要的索引信息提示，用户可以选择具体要打印的条件，在用户选择好条件和检索内容之后，系统就会查询到当前类别文献的所有页面，首先打印显示当前可查询到的所有页面数量，以及下载这些所有的文献需要的大概时间，用户可以手动选择需要打印下载的页数或者具体文献,如图 2 所示。

将爬虫爬取的信息以列表形式记录下来，包括所有的文献的简要信息，比如名称、作者、来源、日期等基本信息，如图 3 所示。

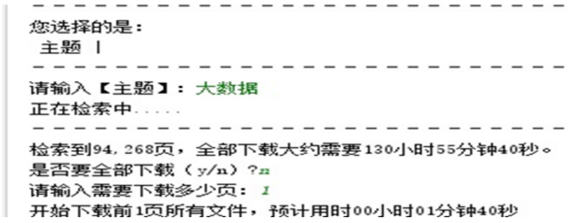

图 2　爬虫检索过程

Fig.2　Reptile retrieval process

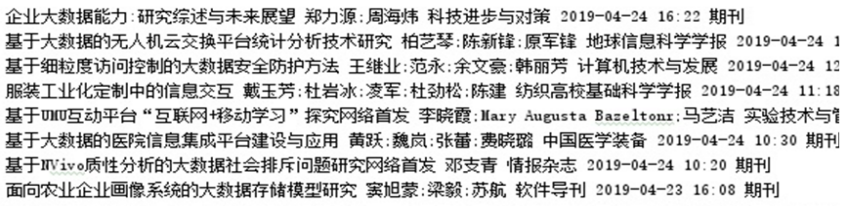

图 3　爬虫文献列表信息

Fig.3　Reptile literature list information

用户在选择好页数或者文献之后，后台则会根据所有选择的条件进行筛选，筛选出所有需要下载的文件数量，然后开始下载，下载的过程中控制台会显示出下载文章的名称，以及具体下载文章的数量，直到所有文章终止，在开始下载的时候，会进行下载的一个时间预判断，给出一个下载的时间范围，然后在下载过程中会设置一个时间读秒，从开始下载到结束下载会设置一个时间计数，统计下载当前所有文章的的实际下载时间，如图 4 所示。

（2）推荐引擎

根据用户的检索过“大数据”历史，还会有相关的推荐列表，如图 5 所示。

目前本文所介绍的系统仅在少量用户中使用，评分数据集较少（20 个用户对 20 个文献共 400 条评分记录）。用均方根误差(RMSE)来计算预测准确度，RMSE 为 25.3%。这是由于测试数据太少引起的，在推广使用后，训练集扩大以后准确率就会上升。

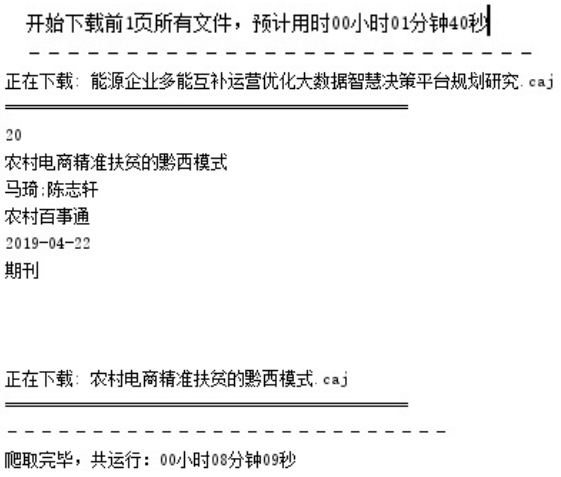

图 4　下载过程

Fig.4　Download process

4　结论

本文一方面使用爬虫技术实现文献的智能检索与关键信息抓取，按需求自动下载文献资源到本地，不需要一篇一篇手工检索下载，提高毕业设计效率。另一方面基于用户协同过滤算法，给学生提供文献推荐列表。经过测试，基于 Python 的文献检索系统能够为学生提供高效准确的文献检索、下载、推荐服务。在下一步的研究过程中将探讨如何将多种推荐算法进行结合，探讨标签与基于内容的混合推荐算法[6]、基于学习风格的自适应推荐算法[7]、用户画像[8]、混合模型推荐算法[9]等提升推荐效率。此外，还要开展用户隐私保护工作[10]。

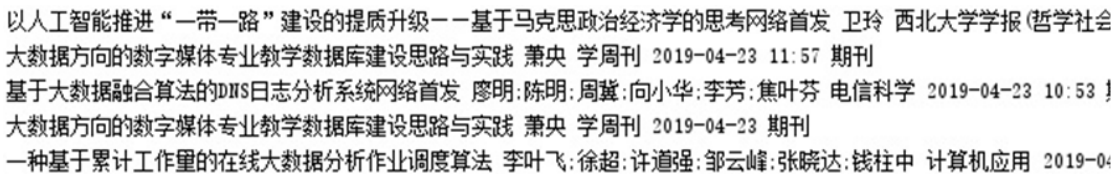

图 5　推荐列表

Fig.5　Recommendation list

参考文献

[1] 杨洋, 李晓风, 赵赫, 等. 基于网络爬虫的文献检索系统的研究和实现[J]. 计算机技术与发展, 2014(11): 35-38.

[2] 刘爱琴, 王友林, 尚珊. 基于爬虫技术的关键词关联推荐算法优化与实现[J]. 情报理论与实践, 2018(4): 134-138.

[3] GOLDBERG D, NICHOLS D, OKI B M, et al. Using collaborative filtering to weave an information tapestry[J]. Communications of the ACM, 1992, 35(12): 61-70.

[4] 阮阳, 刘禹, 韩港成, 等. 基于爬虫的定向数据检索系统[J]. 软件, 2018, 39(5): 118-120.

[5] 项亮. 推荐系统实践[M]. 京: 人民邮电出版社, 2012.

[6] 江周峰, 杨俊, 鄂海红. 结合社会化标签的基于内容的推荐算法[J]. 软件, 2015, 36(1): 1-5.

[7] 许益通, 张冰雪, 赵逢禹. 基于学习风格的自适应学习内容推荐研究[J]. 软件, 2018, 39(4): 01-08.

[8] 李大伟, 杜洪波, 周孝林, 等. 基于“用户画像”挖掘的图书推荐App设计[J]. 软件, 2018, 39(5): 35-37.

[9] 李鹏飞, 吴为民. 基于混合模型推荐算法的优化[J]. 计算机科学, 2014, 41(2): 68-73.

[10] 张小波, 付达杰. 网络信息资源个性化推荐中隐私保护的研究[J]. 软件, 2015, 36(4): 62-66.