哈尔滨工业大学

<<信息检索>>

实验报告

(2022 年度春季学期)

姓名:	王艺丹
学号:	1190201303
学院:	计算机学院
教师:	张宇

实验三 企业搜索系统设计与实现

一、实验目的

本次实验目的是对企业有所系统的设计与实现过程有一个全面的了解。本次实验设计的内容包括：对数据建立索引，实现文档的搜索，并对检索结果排序；实现企业搜索中的分权限访问。

二、实验内容

(一)建立检索系统

对实验 1 获取的 1000 个网页的网页内容建立索引，其次也要对爬取到的所有附件文档建立索引。然后实现一个简单的检索系统，实现数据和文档检索（文档检索要求同学们对文档内容建立索引，可以分开实现，也可以合二为一），并且能够精确的对检索结果进行排序。这一部分要求同学们做成简单的 UI

(二)分权限访问

可以自己定义多种不同的“企业角色”（至少 4 种），这些角色对数据或文档的访问权限不同，然后为每条数据增加访问权限。然后在现有检索系统的基础上加入分权限访问功能，使得不同角色的用户在使用检索系统时，只能看到自己具有访问权限的那部分内容。同时在 3.1 的基础上对应用界面改进，便于切换企业角色。

三、实验过程及结果

综述：

utils.py：实现各功能函数复用

- get_stop_dic(file_path)：构建停用词字典树
- tokenize(content:str)：分词及去停用词处理

BM25.py：类 BM25，自己实现 bm25 检索模型

- cal_scores(self, query)：实现对已分词处理的 query 进行 bm25 得分的计算，利用全 0 向量直接对全部文档进行计算
- get_topk(self, query, k=3)：实现对已分词处理的 query 进行计算检索，返回 bm25 得分最高的 k 个文档对应的索引

RetriverModel.py：类 Retriver，实现对文章与文件的检索

- search_web(self, query)：实现对已分词处理的 query 进行检索，返回前相关性最高的前 5 个文章对应的索引
- search_file(self, query)：实现对已分词处理的 query 进行检索，返回前相关性

最高的前 5 个文件对应的索引

闪光点：

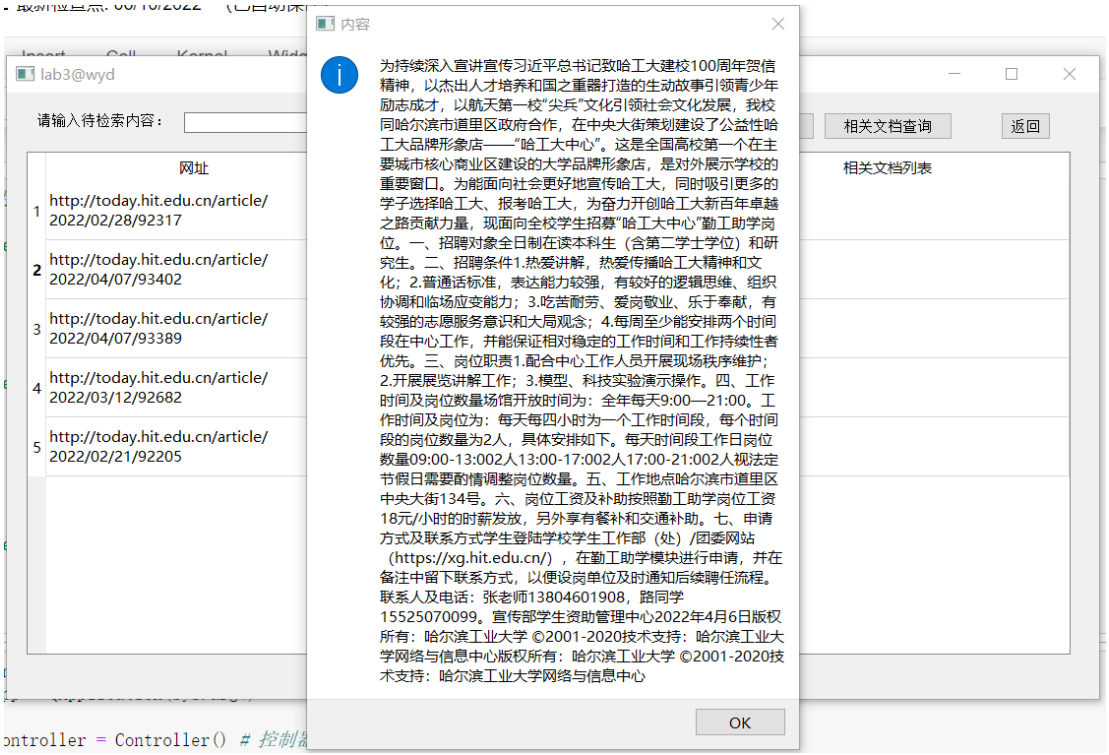
✧ 利用 win32com 与 docx 库实现对文件的读取，表格、文章等信息都可读取
不单单利用文章标题，同时对读取的内容进行处理清洗

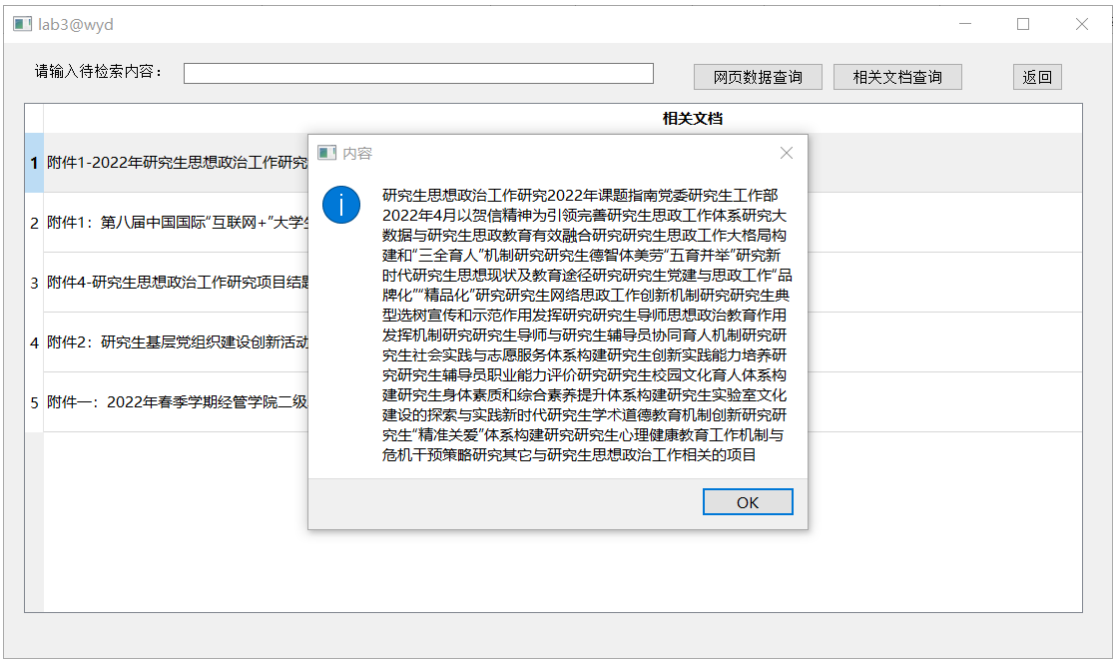
处理附件，读取文件内容用于构建检索模型

```
# 将doc与docx一起处理
def process_doc(file_name):
    pps = []
    doc = word.Documents.Open(FileName=f'{file_dir}{file_name}', Encoding='utf-8-sig')
    try:
        for para in doc.paragraphs:
            pp = ht.clean_text(para.Range.Text)
            pp = repr(pp).replace('\\x07', '')[1:-1].replace('\\r', '').replace(' ','')
            if pp:
                pps.append(pp)
    except:
        print('erro')
    doc.Close()

    content = ''.join(pps)
#     seg = ' '.join(tokenize(content))
    return content
```

✧ 双击检索栏可进行对应内容的查看





3.1 建立检索系统

复用实验 2 实现的 BM25 模型，对文章与文件分别构建一个检索模型，得到最终的 Retriver:

```
1. class Retriver():
2.     def __init__(self, web_dt, file_dt):
3.         self.web_dt = web_dt
4.         self.file_dt = file_dt
5.
6.     def build_model(self, web_model_path, file_model_path):
7.         if os.path.exists(web_model_path):
8.             self.web_model = joblib.load(web_model_path)
9.         else:
10.            self.web_model = BM25(self.web_dt['idx'], self.
   web_dt['paragraphs'])
11.            joblib.dump(self.web_model, web_model_path)
12.
13.        if os.path.exists(file_model_path):
14.            self.file_model = joblib.load(file_model_path)
15.        else:
16.            self.file_model = BM25(self.file_dt['idx'], sel
   f.file_dt['content'])
17.            joblib.dump(self.file_model, file_model_path)
18.
19.    def search_web(self, query):
```

```
20.         query = ' '.join(tokenize(query)) # 对query 分词并且去停用词
21.         topk = self.web_model.get_topk(query, k=5)
22.         return topk
23.
24.     def search_file(self, query):
25.         query = ' '.join(tokenize(query)) # 对query 分词并且去停用词
26.         topk = self.file_model.get_topk(query, k=5)
27.         return topk
```

3.2 分权限访问

该部分内容实现为对检索结果进行过滤：检索结果的后处理，过滤掉用户不具备访问权限的结果；用户等级如下 4 个：

老师：level=1；可获取相关性前 5 的数据, url:可访问，title:可访问，文档:可访问, paragraph:可访问

学生：level=2；可获取相关性前 5 的数据, url:可访问，title:可访问，文档:禁止访问, paragraph:可访问

家长：level=3；可获取相关性前 3 的数据, url:禁止访问，title:可访问，文档:禁止访问, paragraph:可访问

游客：level=4；可获取相关性前 1 的数据, url:禁止访问，title:可访问，文档:禁止访问, paragraph:可访问

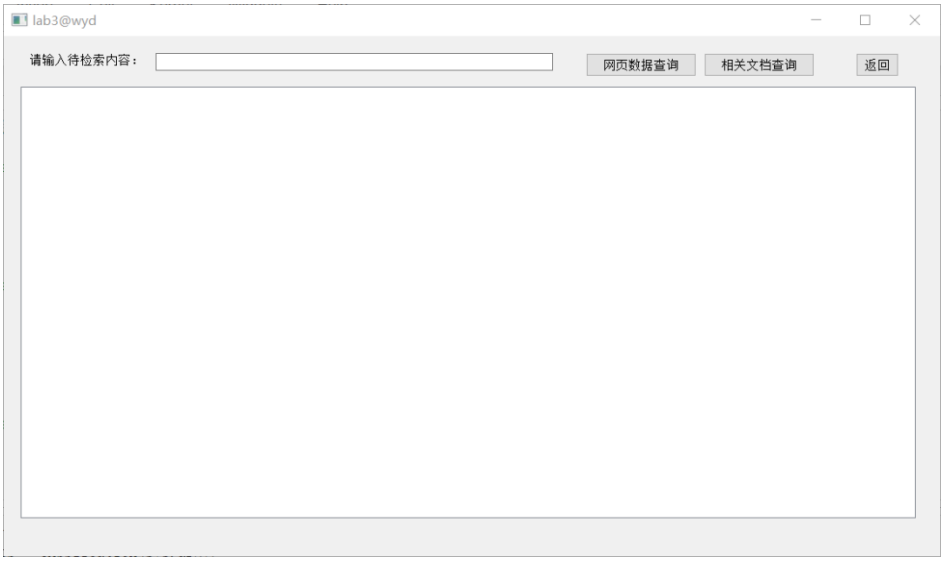
UI 的实现：

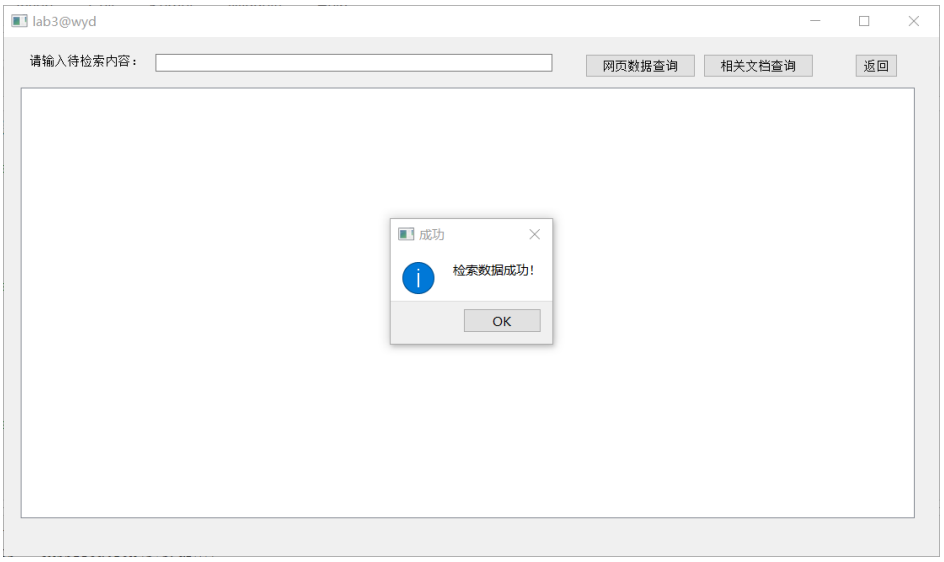
基于 pyqt5 实现，对结果以 TableView 的格式进行展示，最初界面为用户登录界面，登录后跳转至检索页面，登录时判断用户身份及等级

实验结果展示：

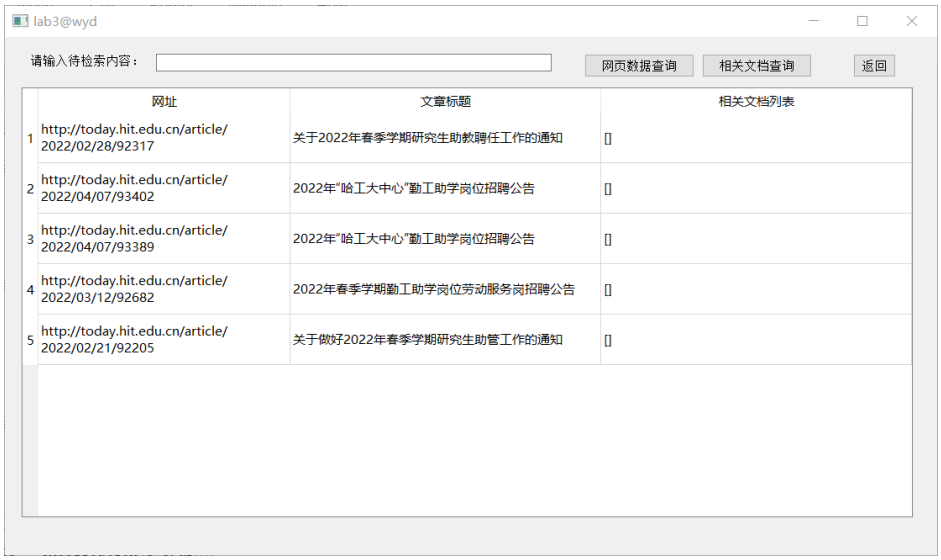
系统登录页面：

系统检索页面：

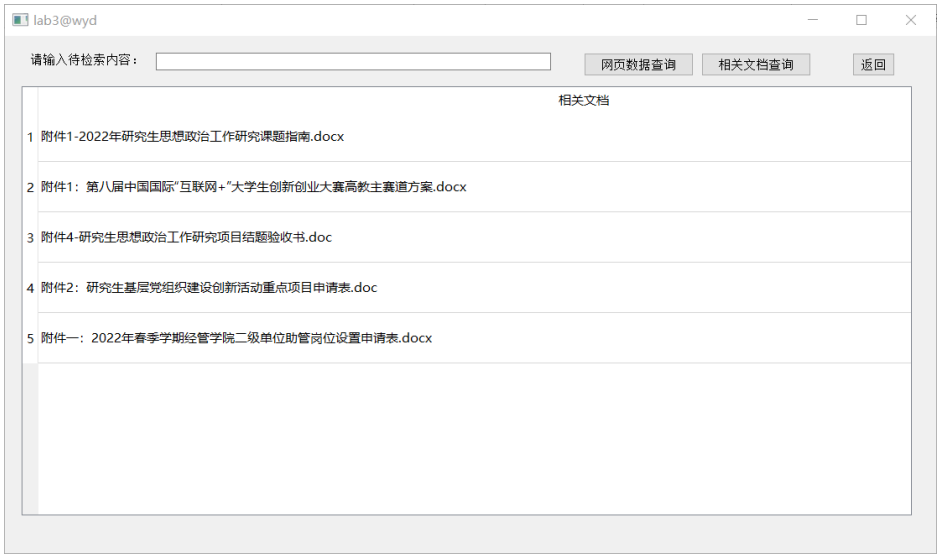




数据检索：


lab3@wyd
请输入待检索内容：
网页数据查询
相关文档查询
返回
网址
文章标题
相关文档列表
1 http://today.hit.edu.cn/article/2022/02/28/92317 关于2022年春季学期研究生助教聘任工作的通知 []
2 http://today.hit.edu.cn/article/2022/04/07/93402 2022年"哈工大中心"勤工助学岗位招聘公告 []
3 http://today.hit.edu.cn/article/2022/04/07/93389 2022年"哈工大中心"勤工助学岗位招聘公告 []
4 http://today.hit.edu.cn/article/2022/03/12/92682 2022年春季学期勤工助学岗位劳动服务岗招聘公告 []
5 http://today.hit.edu.cn/article/2022/02/21/92205 关于做好2022年春季学期研究生助管工作的通知 []

文档检索：


lab3@wyd
请输入待检索内容：
网页数据查询
相关文档查询
返回
相关文档
1 附件1-2022年研究生思想政治工作研究课题指南.docx
2 附件1：第八届中国国际"互联网+"大学生创新创业大赛高教主赛道方案.docx
3 附件4-研究生思想政治工作研究项目结题验收书.doc
4 附件2：研究生基层党组织建设创新活动重点项目申请表.doc
5 附件一：2022年春季学期经管学院二级单位助管岗位设置申请表.docx

实验中的问题：

- **如何实现页面跳转**

  阅相关资料得知可以利用 pyqtSignal 实现页面跳转，在不同窗口之间设置跳转参数，并与对应的跳转函数绑定，提交对应的跳转信号，利用一个 Controller 控制页面跳转

  switch_window = pyqtSignal() # 界面跳转信号

  self.switch_window.emit() # 页面跳转

```
# 利用控制器来控制页面的跳转
class Controller:
    def __init__(self):
        self.login = MainWindow()
        self.search = SearchWindow()
        pass

    # login 窗口
    def show_login(self):
        self.search.close()
        self.login.switch_window.connect(self.show_search) # 跳转信号绑定
        self.login.level_info.connect(self.search.get_data) # 参数传递信号绑定
        self.login.show()

    # 跳转到 search 窗口，并关闭原页面
    def show_search(self):
        self.login.close()
        self.search.switch_window.connect(self.show_login) # 跳转信号绑定
        self.search.show()
```

- **如何将登录页面的身份信息传递给检索页面，以便根据身份对结果进行过滤**

  查阅相关资料得知可以利用 pyqtSignal 函数也可以进行页面间参数的传递；再跳转后的窗口与信号绑定相应的设置参数的函数，即可完成页面间的参数传递

```
class MainWindow(QMainWindow, login_Form):

    switch_window = pyqtSignal() # 界面跳转信号
    level_info = pyqtSignal(int) # 页面间传递参数
```

```
self.level_info.emit(level) # 传递level信息
self.switch_window.emit() # 页面跳转
```

```
def get_data(self, level_info):
    self.level = level_info  # 接受 login 页面传过来的level_info，并保存
```

四、实验心得

- 学会了 PyQt5 以及 designer 的使用，实现简易的 UI
- 学会了使用 python 对文档文件的读取处理
- 对企业搜索系统的简易框架有了认识与实践