# 哈尔滨工业大学

# <<信息检索>>

# 实验报告

(2021 年度春季学期)

| | |
|---|---|
| 姓名： | 刘璟烁 |
| 学号： | |
| 学院： | |
| 教师： | |

# 实验三 企业搜索系统的设计与实现

## 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

本次实验的内容主要分为两部分，包括建立检索系统和分权限访问:

1. 建立检索系统

任务描述:本次实验使用到的数据是实验 1 中爬取的网页数据。首先要对 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。这一部分要求做成简单的 UI 应用或者是 Web 应用，来保证易用性。

2. 分权限访问

任务描述: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。同时在第一部分的基础上对应用界面改进，便于切换企业角色。分权限访问的实现一般有以下两种模式:1. 检索条件控制:对检索条件加以限制，仅在用户访问权限内的文档中检索。 2. 检索结果过滤:检索结果的后处理，过滤掉用户不具备访问权限的结果。

## 三、实验过程及结果

1. 建立检索系统

**关键步骤实现：**

该部分的实现首先分别对网页文本内容和附件内容建立倒排索引。

在 build_index.py 文件中，实现了对网页和附件文本内容的预处理：首先将文本内容进行分句、分词，在分词的过程中逐词构建倒排索引；在这里，对于存储于 preprocessed.json 中的网页文本内容，我将网页内容在该 json 文件中的行号作为该网页的 pid 号，用于构建倒排索引；而对于所有的 docx 附件，我们将所有 docx 附件的文件名存储入一个全局存储结构中，逐个为其分配 pid 并根据此信息进行倒排索引的构建。

随后进行检索系统的构建，本次实验使用了基于 bm25 算法的检索系统。

对于检索系统的具体实现，分别使用网页文本内容和附件文档的文本内容两个语料库作为训练预料构建了两个 bm25 检索模型；以其中的网页文本检索系统为例，对于一个 query 查询，首先调用上一部分所述的网页文本的倒排索引进行相关文档的检索，随后使用网页文本对应的 bm25 检索系统对这些相关文档进行相似度分数排序，并按照相似度降序依次返回结果；对于附件内容检索系统，其查询算法与前者基本一致。构建的检索系统以 GUI 的方式呈现，请见于实验结果截图部分。

2. 分权限访问

**关键步骤实现：**

这一部分要求对使用企业检索系统的用户设置权限访问功能，对于四种身份的使用者，我们对检索系统返回的检索结果进行过滤，即过滤掉用户不具备访问权限的结果。

具体的实现方法为，设置“董事长”、“总经理”、“部门经理”、“员工”四种用户身份。以对网页内容的检索系统为例，赋予董事长所有文档（约 1100 个）的使用权限，总经理拥有 pid 为前 500 的文档的访问权限，以此类推，员工只有 pid 前 100 的文档的访问权限；而对于附件的检索系统，本次实现按照附件文档数的四分之一的倍数分别赋予上述四种用户身份访问权限。在向用户界面返回结果时，根据文档对应的 pid 和用户身份进行判断是否有权访问，若否则不在用户界面上返回该内容。

**实验结果截图**（包含检索系统建立和分权限访问两部分内容）：

在检索系统实现部分，为网页文本构建的倒排索引文档(web_inverted_index.txt)结果部分截图如下，每个词项后跟随包含该词项的文档列表：

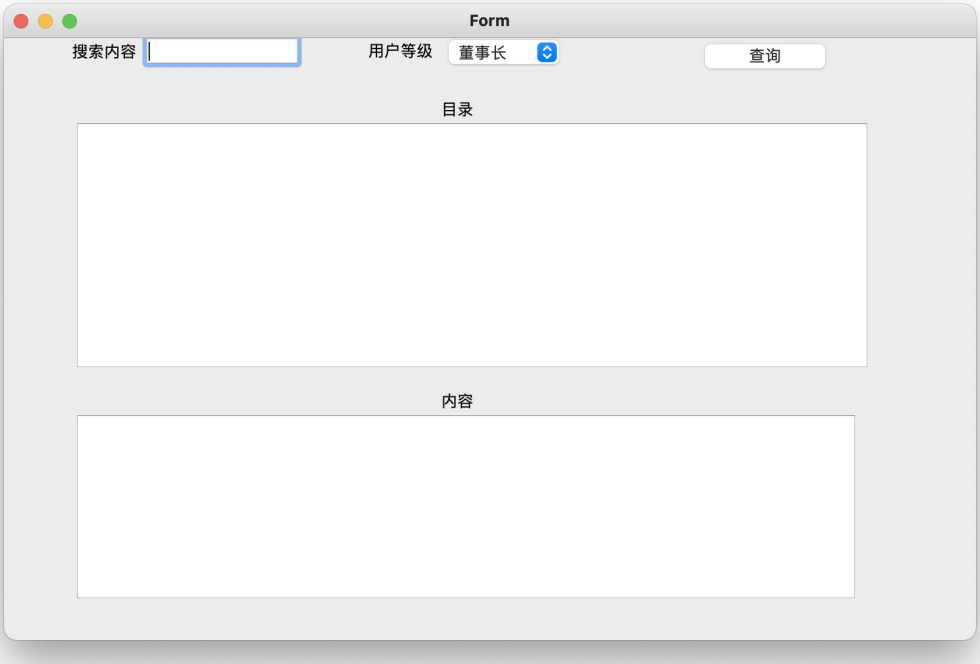
```
各学科 0 2 515 546 421 583 457 652 915 372
制定 0 896 2 515 392 520 150 790 25 921 539 926 546 677 425 430 432 946 820 3
博士生 0 2 390 523 142 15 143 18 402 530 531 150 786 28 417 418 35 546 425 42
导师 0 640 2 515 901 780 142 143 145 18 402 22 156 418 546 425 426 427 939 43
招生 0 128 2 515 520 782 143 530 536 797 798 799 926 546 802 804 422 809 432
计划 0 2 515 4 516 520 527 21 25 539 546 564 54 566 568 57 569 69 582 71 76 5
审核 0 2 770 260 772 784 792 794 796 799 546 425 43 175 304 815 567 56 57 58
标准 0 896 2 515 260 520 266 907 910 912 786 914 915 662 279 539 798 926 927
的 0 2 3 4 6 8 11 14 15 16 18 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35
原则 0 2 515 899 520 534 279 790 539 796 797 926 927 546 931 804 298 304 817
```

为附件文本内容构建倒排索引文档(docx_inverted_index.txt)结果部分截图如下：

```
研究生 1 10 11 24 25
学术交流 1
活动 1 6 11 12 14
管理 1 3 9 11 14 21 24 25
暂行办法 1
```

下面展示检索系统的 GUI 界面，首先进入开始界面选择进行网页内容检索或者附件文档内容检索：

我们首先进行网页文本内容的检索，点击左侧“网页内容检索”，进入如下界面：



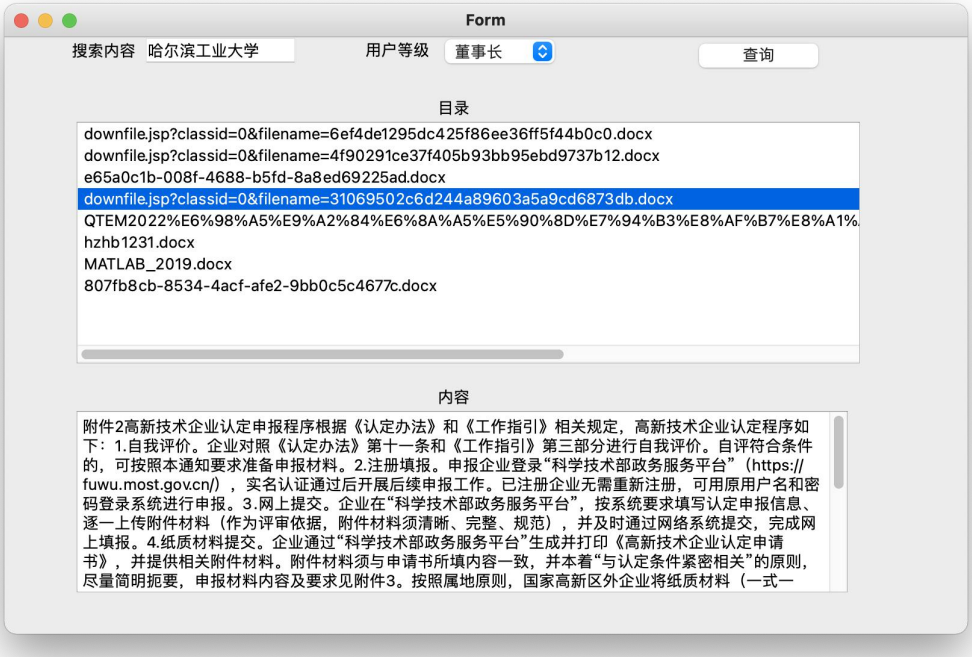

使用董事长身份查询，例如输入“党的建设”，可以看到返回相关若干条网页的标题，任意点击其中一个标题，将在内容栏显示正文内容：


使用员工身份查询相同的内容，可以发现仅有一条有权访问的网页，点击后将在

内容一栏返回网页正文：


随后展示对于附件文本内容的检索，对于特定内容的检索将返回相关附件的文件名列表，点击某一文件名将会在内容栏显示附件中的内容：

使用董事长身份，点击查询“哈尔滨工业大学”，得到附件文件名列表如下，任意点击其中一个附件，“内容”一栏将返回该附件中的正文：



使用员工身份，查询相同内容，只看到了一条相关的有权访问的附件，点击查看内容如下：


## 四、实验心得

**遇到的问题及解决方法：**

问题一：在检索系统的实现过程中，需要分别为网页正文和附件正文建立倒排索引，而对于实验一中保存的 preprocessed.json 中的网页内容和 files 文档中保存的附件内容，并没有建立相应的文档 id 索引，导致影响到了倒排索引的建立。

解决：为了解决这一问题，对于 preprocessed.json 中的网页内容，可以按照网页对应的行号作为该网页的 pid，而对于 docx 附件则可以逐个分配 pid，从而解决倒排索引构建的问题。

问题二：在分权限访问实现的过程中，需要根据用户身份过滤特定的文本内容，但由于难以分析自行爬取的内容的重要程度，因此难以决定具体的过滤内容。

解决：对于问题二，本次实验简化的将文本对应的 pid 作为衡量其重要程度的标准，例如最低权限的用户只能访问 pid 值小于 100 的文本，而最高权限用户可以

访问所有的内容。

**实验收获：**

本次实验实现了一个简洁的企业检索系统，通过对企业中数据的预处理、以及企业检索系统的建立，加深了对课程内容中索引部分以及检索系统模型的进一步理解；同时通过具体的为访问用户设立权限，以及根据用户权限对相应结果进行过滤，对现代企业检索系统的需求有了更深刻的体悟。