****

**智能信息检索(python)**

**实验报告**

**实验题目： 设计实现智能信息检索系统**

**专 业： 人工智能**

**姓 名： 马承乾**

**实验日期**：  **2023.7.11**

**指导教师： 林达真**

1. **实验目的**

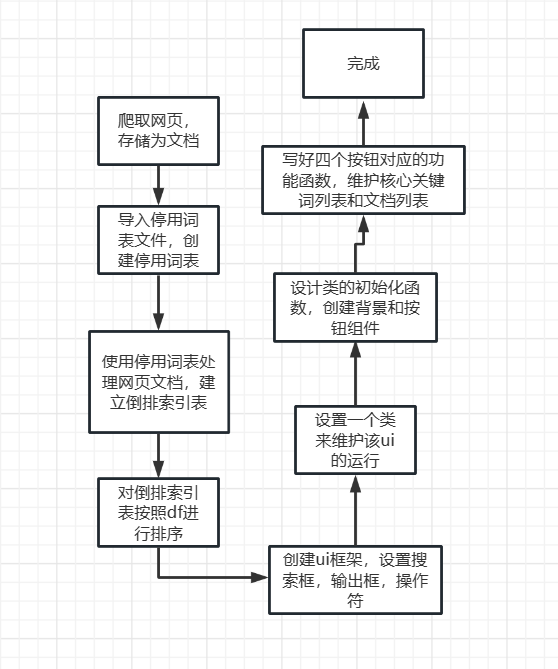
建立一个能实现布尔检索功能的检索系统。

重点：ui界面，网页排序，多关键字检索。

1. **实验内容**
2. 采集网址包含xmu.edu.cn的所有静态主页
3. 抽取网页中的文本内容，每个网页一个文档。
4. 数据预处理：
   1. 分词（结巴分词）
   2. 去除非汉字数据（数字、英文字符、标点符号……）
   3. 去除停用词
5. 建立独立的倒排索引表
6. 实现简单的布尔查询，对采集到的网页内容进行检索，支持多关键词AND，OR和NOT

附加：设计一个排序的方法，对布尔检索的AND查询得到文档进行排序

1. **实验步骤以及结果（完整代码见附件）**
2. 流程总览:

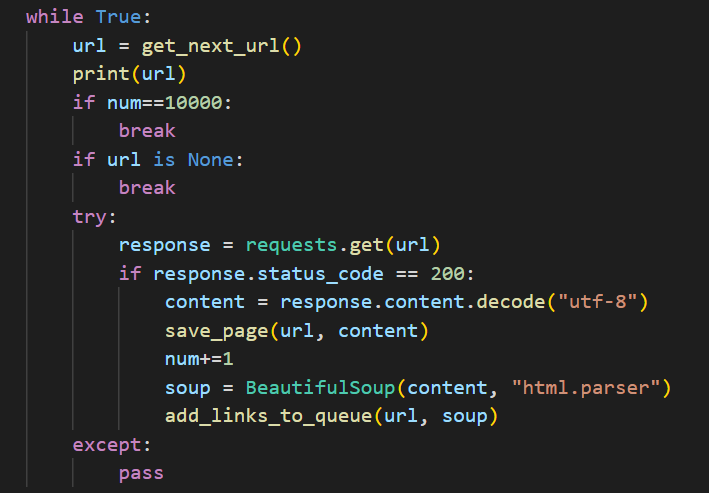


1. 检索系统使用方式，在搜索框中输入关键字，点击display result即可查看结果；

如果要查看当前关键字与关键字A与的结果，在输入框中输入A后再点击AND即可查看结果；使用or,not类似；

若要清空当前关键字，进行新的查询，直接在输入框中输入并且display result，或者先点击clear result之后再进行输入即可。

1. 爬取网页，存储为文档：

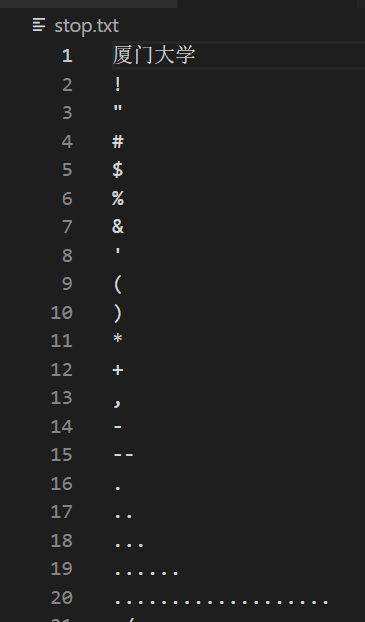


用crawl.py进行爬取，设置一个列表page\_pkg和对应的下标pkg\_index用来存放和取出网址，get\_next\_url的功能是从列表中取出新的网址，并将其输出给用户进行提示；

成功访问网页以后用函数save\_page将网页储存为文件，存储静态网页对应的文本，随后在新提取的静态网页中取出新的链接，用add\_links\_to\_queue放入page\_pkg，在放入列表的过程中注意去重，不要重复爬取相同网站。

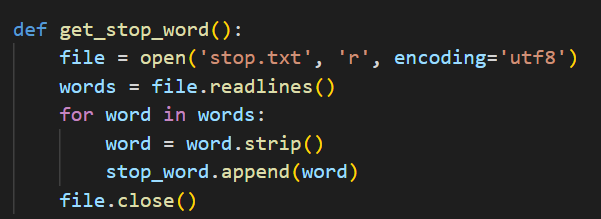
1. 导入停用词表文件，创建停用词表

下载现有的停用词表，导入后创建停用词表。



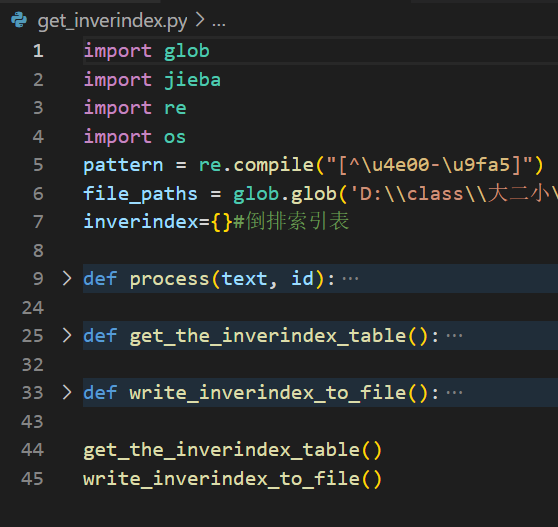
注意，由于本检索系统基于厦门大学网站，所以为了减少倒排索引表的负担，将厦门大学写入停用词表。

在work.py文件中，导入该文件中的停用词表，得到停用词表stop\_word：

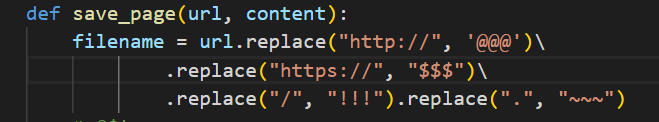


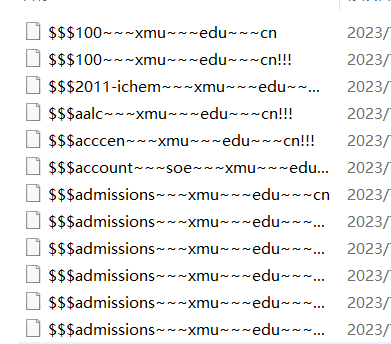
1. 使用停用词表处理网页文档，建立倒排索引表

在正式开始工作之前，用get\_inverindex.py创建倒排索引表：

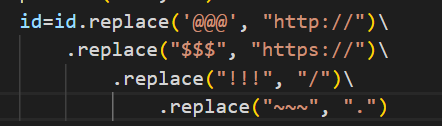


在创建倒排所应表的过程中需要注意，按照作业要求只保留中文信息，同时由于对应网址中有//等不允许在文件名中出现的特殊符号，所以要对这些符号进行特殊处理；在爬虫中，我们将文件名中的这些特殊符号进行了转换：

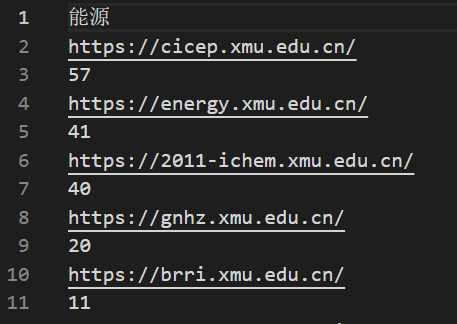




在创建倒排索引表过程中，我们进行逆转换



建立的倒排索引表是一个字典， 字典的键是关键字K，值是很多字典，每个字典中，键A是文件名，值B是该K在A中出现的次数。进而我们创建了倒排索引表且将之写入了文件：



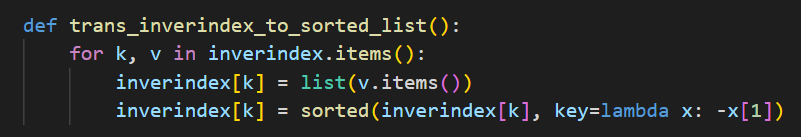
这样在work.py中我们就可以对其进行使用：



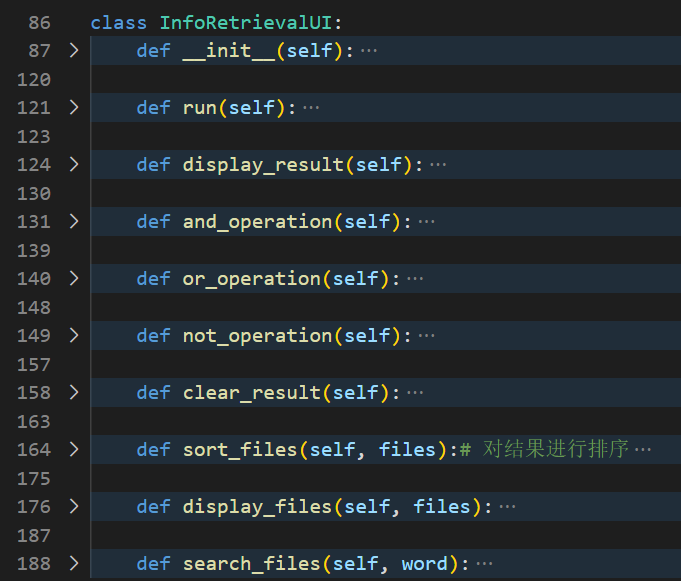
使用work.py中的这两个函数读取文件并且将倒排索引表转化为有序的形式。

1. 对倒排索引表按照df进行排序

对倒排索引表按照df进行排序，在inverindex这个字典中，键K是关键字，值是很多二元列表，每个二元列表中有两个元素A,B，其中A是文件名，B是K在A中出现的次数，出现次数越多越靠前。

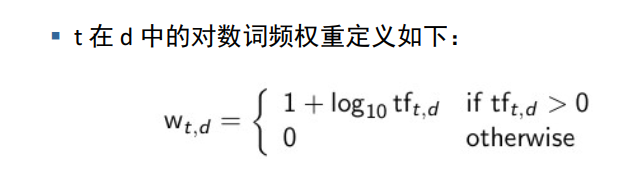


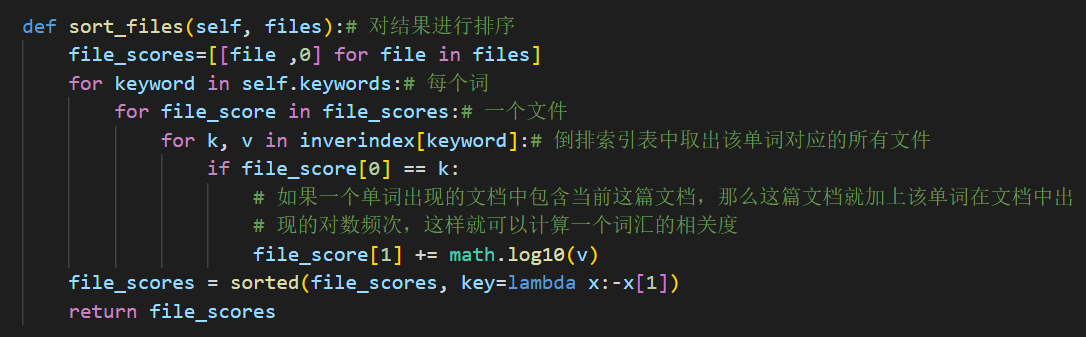
1. 创建ui框架，设置搜索框，输出框，操作符设置一个类来维护该ui的运行



and\_opertaion,or\_operation,not\_operration分别是三种操作符的按钮,display是显示结果的函数，clear\_result对现有的关键词和文件检索结果清空,sort\_files在输出之前进行排序。

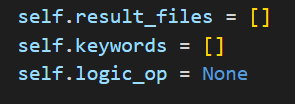
排序依据是对数词频：



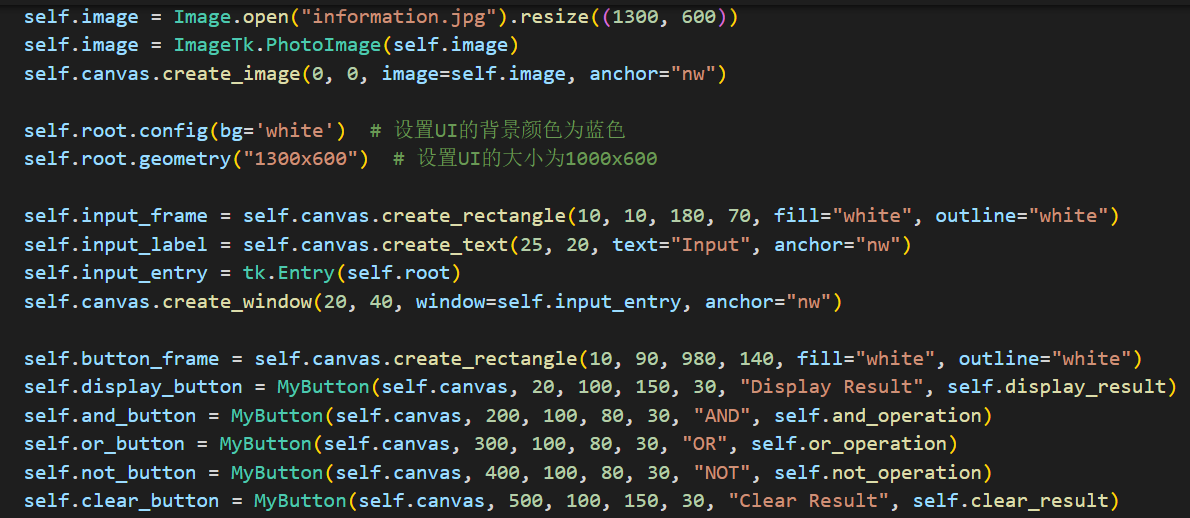


如果一个单词出现的文档中包含当前这篇文档，那么这篇文档就加上该单词在文档中出现的对数频次，这样就可以计算一个词汇的相关度。

在该类的初始化函数中，维护一个result\_files来存储已经检索到的文件结果，keywords存储当前的关键字，logic\_op存储当前的逻辑操作符。这样就可以进行多个关键字的处理。

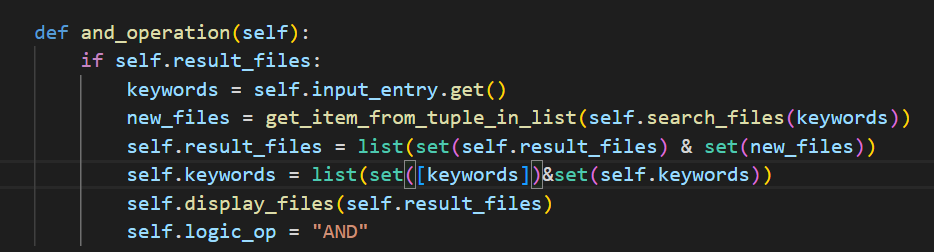


1. 设计类的初始化函数，创建背景和按钮组件



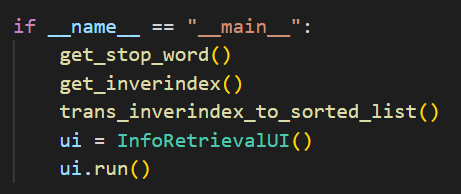
使用tkinter中的组件功能，并设置个性化图片背景，使得该界面尽量美观。

1. 写好四个按钮对应的功能函数，维护核心关键词列表和文档列表



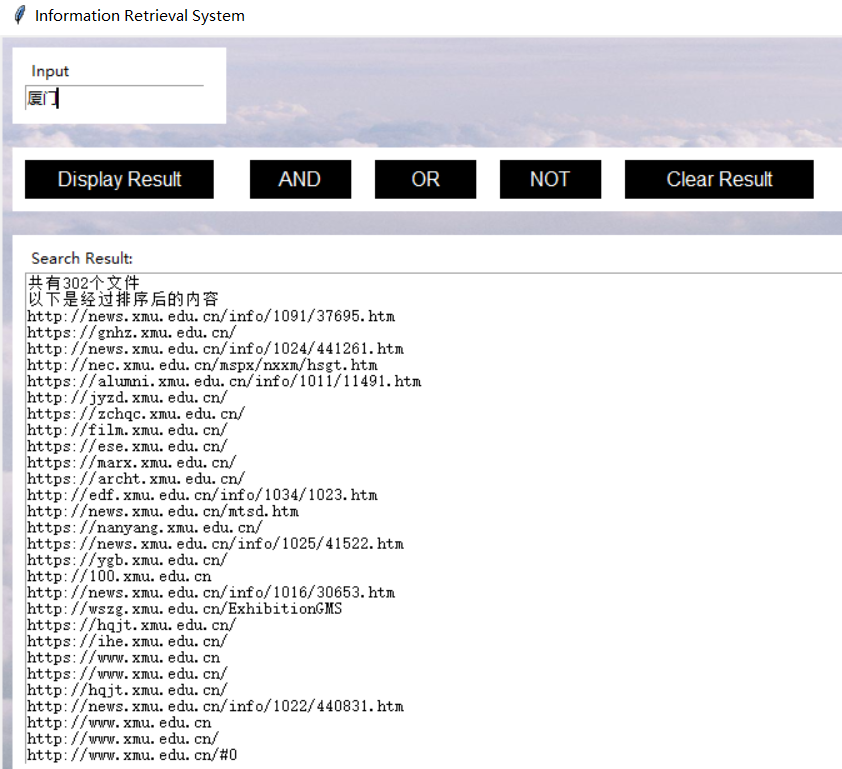
另外两个操作符类似，这里用好set类的特性，和被重构的&操作符，进行简单的逻辑操作。

1. 使用封装好的函数进行初始化以后，实例化并运行该我们设计好的ui界面：



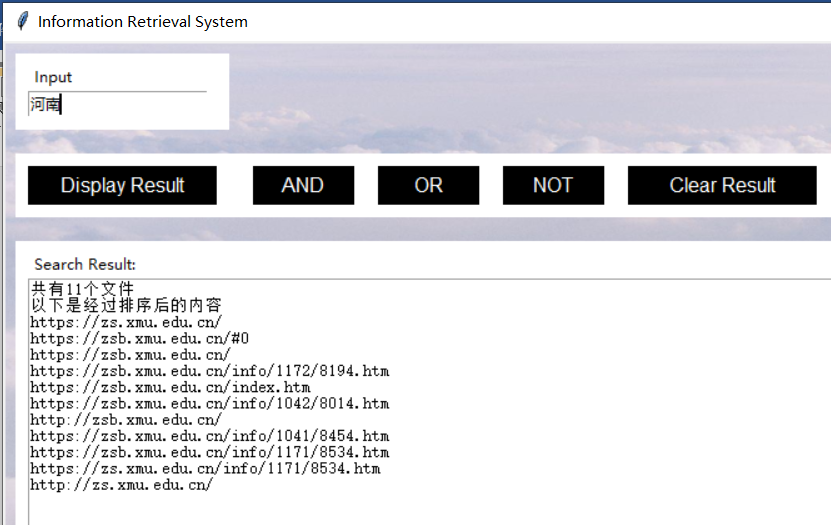
1. **实验结果与分析**
2. 进行单个关键词进行检索：

关键词：厦门



1. 进行多关键词与操作：

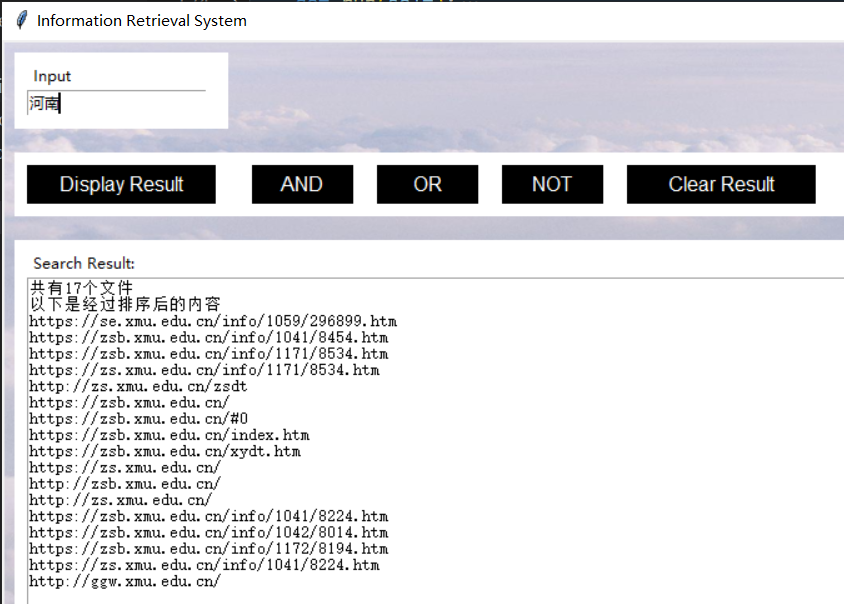
关键词：厦门&大学&河南

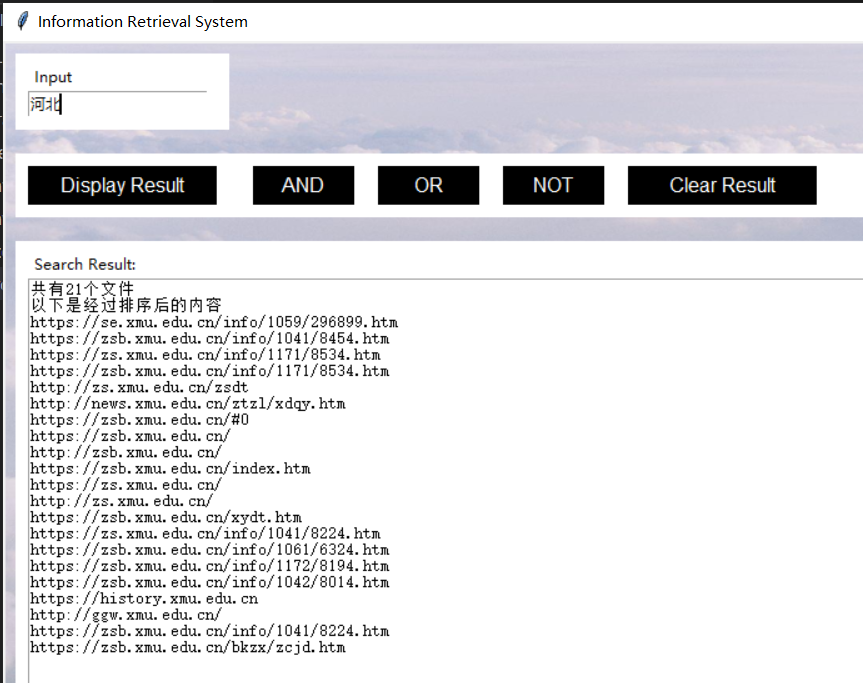


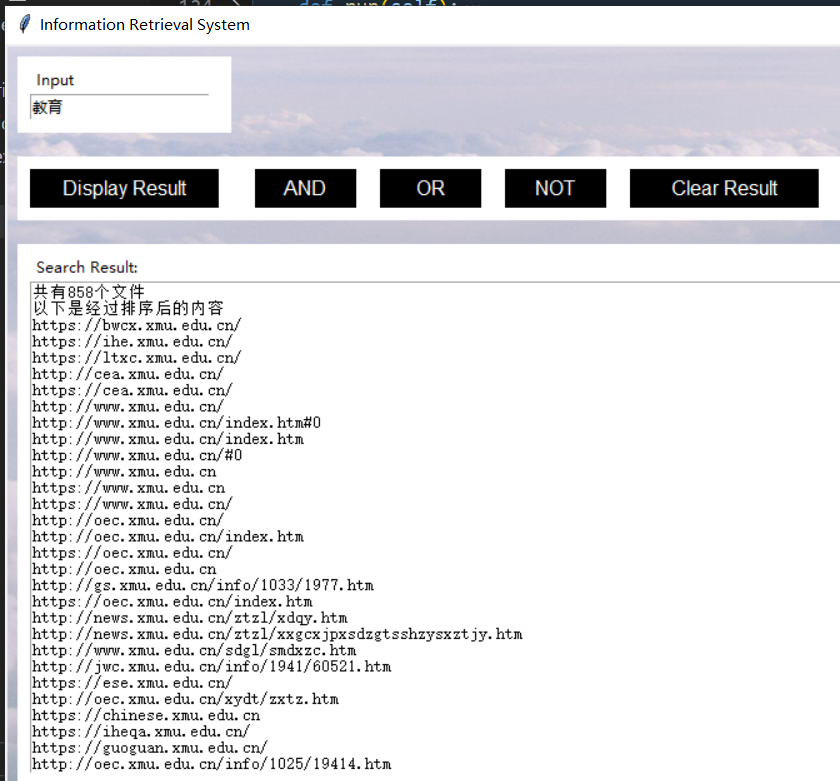
显然检索结果变少了，符合布尔检索的规律。

1. 多关键词或操作：

关键词：河南 or 河北 or 教育



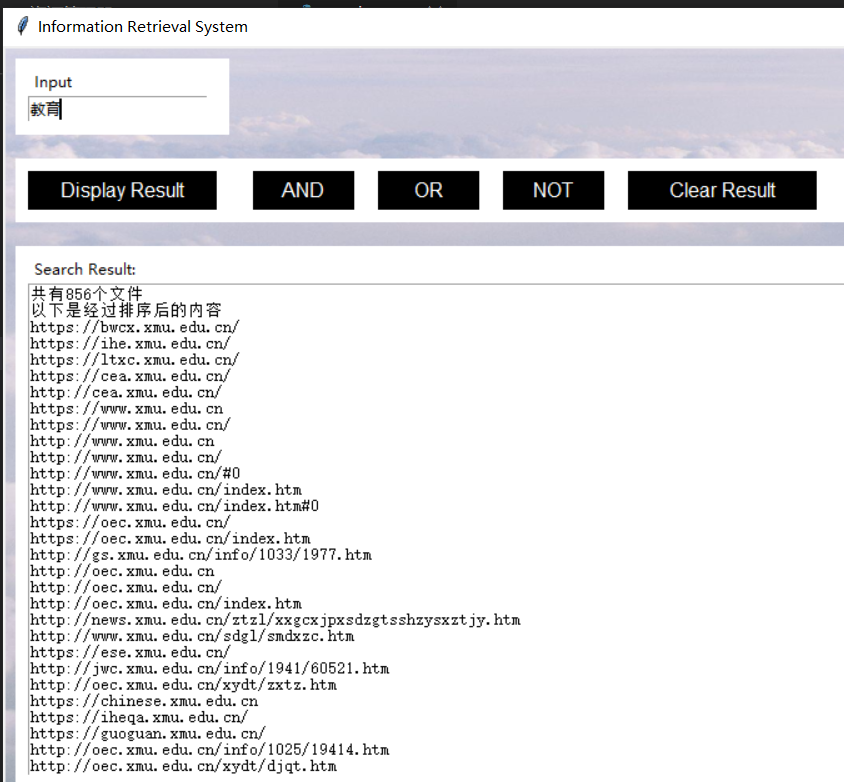


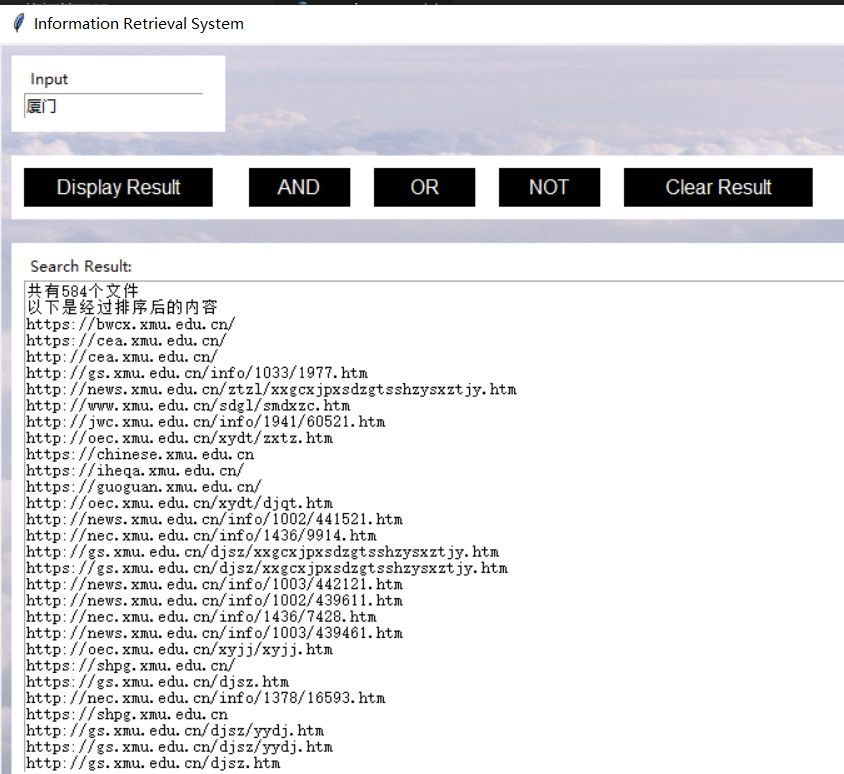


检索结果不断增多，符合逻辑运算的逻辑。

1. 多关键词非操作：

教育 not 厦门

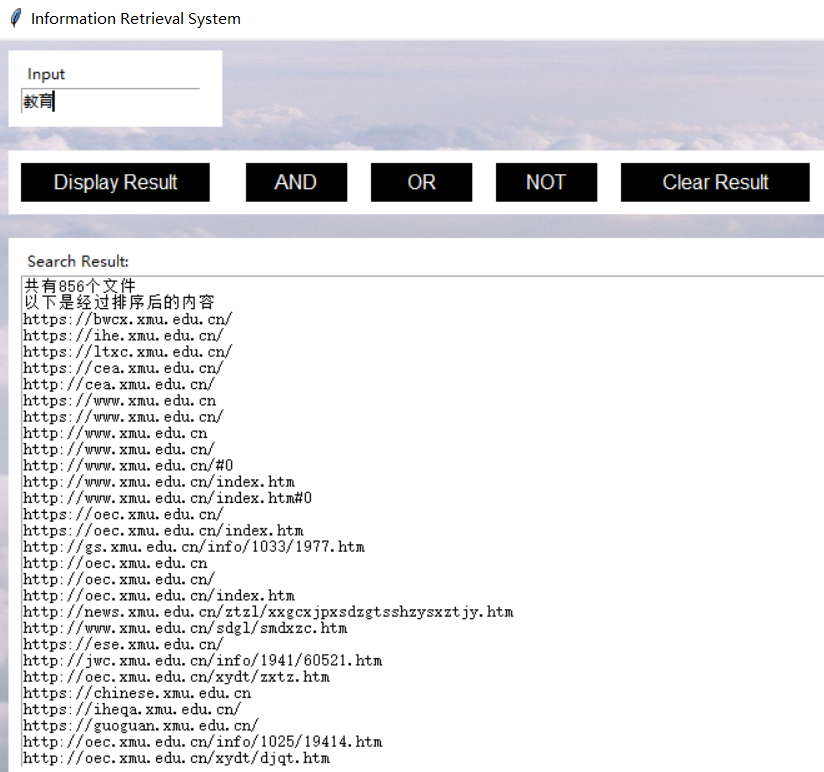


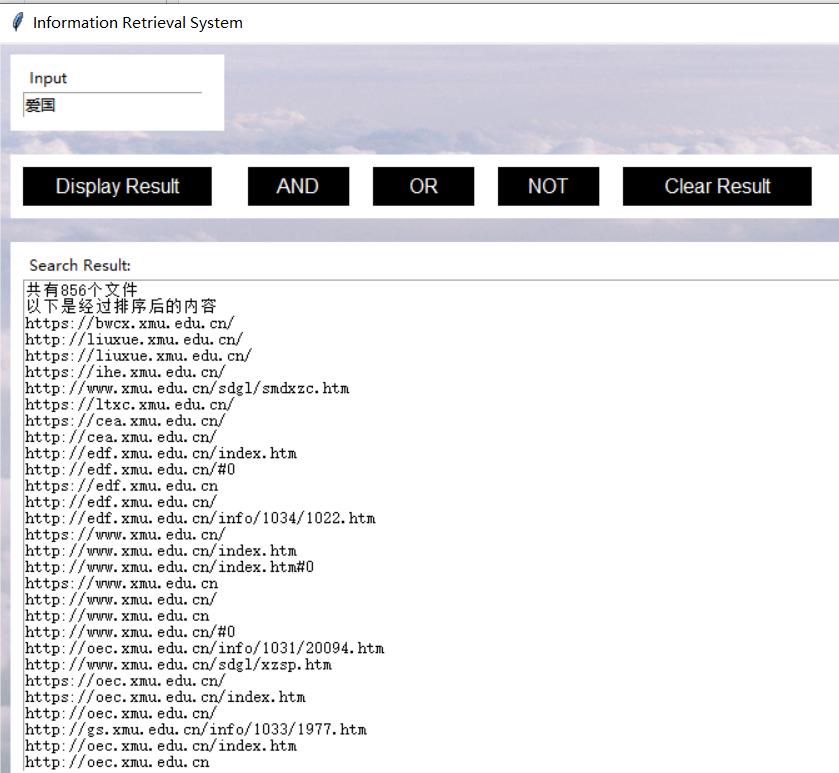


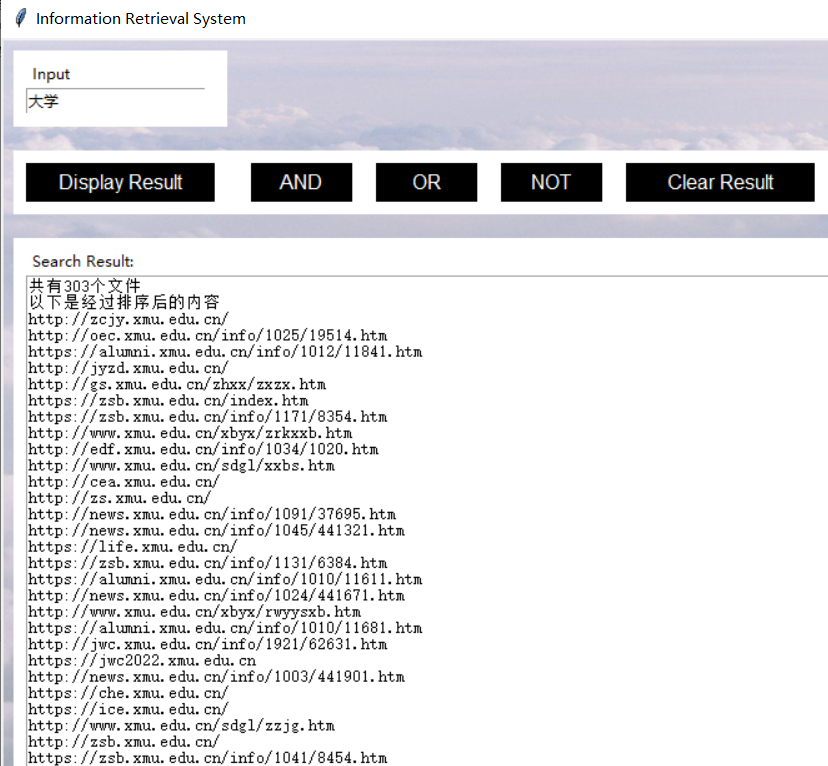
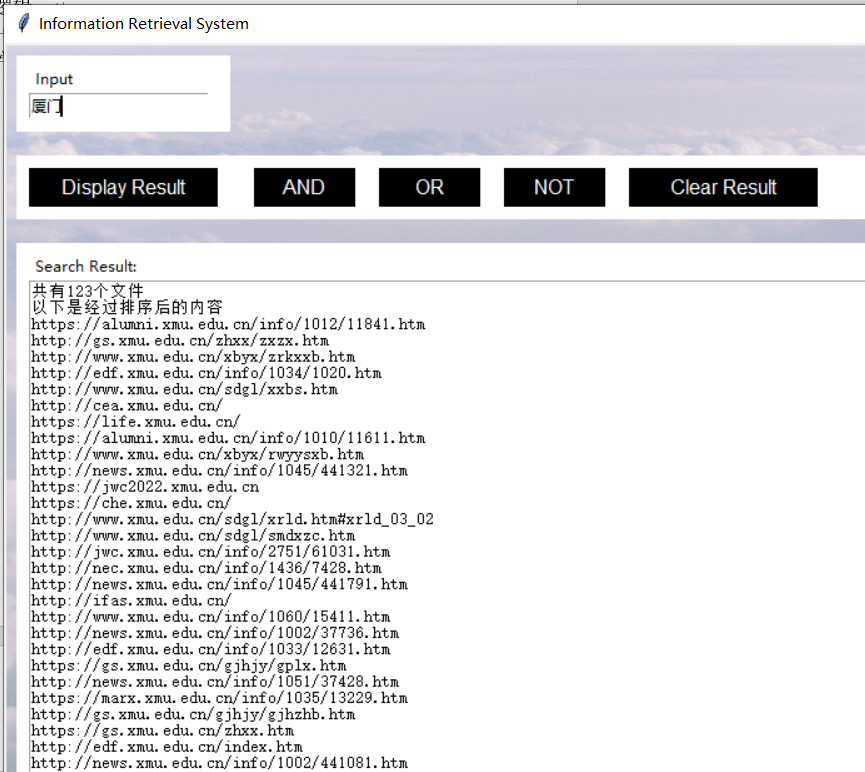
结果不断变少，符合逻辑。

1. 多关键词混合操作：

教育or 爱国 and 大学 not 厦门





1. **实验总结**

通过本次实验，我对信息检索技术有了更深刻的认识，尤其是使用类封装以后，可以让代码变得非常结构清晰和明了。

在检索过程中，建立起有序的倒排索引表可以避免在布尔检索时反复遍历所有文档，从而大大节省时间。排序意味着需要对文件和关键字之间的相关度进行评判，评判标准的质量决定了布尔检索的质量。

在实验过程之中，我使用到了tkinter以及JIEBA等现有的python函数库，这些库可以在很大程度上简化我们的工作，让我们的精力可以更好的放到重点的主干上。