中南大學
CENTRAL SOUTH UNIVERSITY

《智能搜索引擎》
实验报告

学生姓名	卜德华
学　　院	计算机学院
专业班级	大数据 2201 班
老　　师	刘嫔
学　　号	8208220314

2025 年 5 月 29 日

目录

一、实验目的

1. 掌握搜索引擎的设计理念和工作原理，学会设计和实现一个搜索引擎系统，提高实际动手能力，为具备一定的设计解决方案以解决复杂工程问题的能力打下坚实的基础。
2. 通过课程实践，建立创新能力通过课外导学的模式，提升自主学习和终身学习的意识，形成不断学习和适应发展素质。
3. 学会组内合作、沟通交流共同开发、调试程序。

二、实验背景

随着互联网信息爆炸式增长，用户对海量数据中精准、高效地获取目标内容提出了更高的要求。搜索引擎技术作为支撑现代信息获取的核心手段，其基本流程包括网页抓取、信息抽取、内容索引与检索排序等多个关键模块。其中，倒排索引作为搜索引擎的核心数据结构，能够将非结构化文本高效转换为可查询的结构化索引，是实现快速关键词检索与相关性排序的基础。

在中文环境下，搜索引擎构建面临更加复杂的挑战，如中文分词、词义歧义、多义短语识别等问题。因此，如何结合中文分词技术（如 jieba）构建适用于中文图书数据的倒排索引系统，并基于用户查询完成相关结果的召回与排序，是本实验的重要课题之一。

三、实验原理

本实验围绕搜索引擎的索引与检索模块展开，主要涉及以下四个核心技术原理：

1. 网页爬虫原理

网页爬虫（Web Crawler）是搜索引擎的入口模块，其基本原理是模拟浏览器

发送 HTTP 请求，下载网页 HTML 内容，并通过链接解析算法提取页面中的超链接，递归抓取新的页面。整个过程通常包含以下几个关键机制：

1. 请求调度：通过队列（如 BFS）管理待抓取 URL，防止重复抓取；
2. 反爬策略应对：设置合理的请求头（如 User-Agent）、延时机制（Sleep）避免触发网站封禁；
3. 网页下载与存储：使用 requests 获取网页内容，配合 BeautifulSoup 进行 DOM 树解析，并以 .html 原始格式保存用于后续处理；
4. 链接提取与规范化：解析 <a href> 标签中的链接，使用 urljoin 拼接相对路径，并限定爬取范围在同一域名下。

通过爬虫模块可获取结构完整、内容真实的网页数据，为后续的内容提取和索引构建奠定基础。

2. 网页内容提取原理

网页内容通常包含大量冗余信息（导航栏、广告、脚注等），为了提取有价值的正文信息，需要对 HTML 页面进行语义结构分析。本实验以图书类网站为对象，提取 <article class="product_pod"> 等结构化块中的字段内容，主要包括：

1. 标题提取（title）：位于 <h3> 标签内，通过 title 属性或文本节点获取；
2. 价格提取（price）：通过 CSS 类名 .price_color 定位；
3. 库存状态（availability）：提取 .availability 节点中的文本；
4. 评分提取（rating）：通过标签类名（如 star-rating Three）解析出星级；
5. 商品链接（url）：从 <a> 标签的 href 属性获取并标准化为绝对路径。

提取结果被统一保存为 JSONL 格式，便于后续构建倒排索引与语义分析。

3. 倒排索引库原理

倒排索引（Inverted Index）是现代搜索引擎的核心数据结构，它将“文档到词”的正向映射，转换为“词到文档”的逆向映射，能够显著提升检索效率。

其基本结构为一个字典 Index = { term → [doc_id1, doc_id2, ...] }，具体原理如下：

1. 构建过程：
   - 对每条文档（本实验中为图书记录的标题）进行分词；
   - 遍历每个词条，将该词条出现的文档 ID 加入对应 posting list；
2. 查询过程：
   - 将用户查询语句分词后，在倒排表中查找所有包含该词的文档列表；
   - 若为多词查询（AND 模式），取 posting list 的交集；OR 模式则取并集；
3. 相关性评分：
   - 初始版本以“命中词数”作为简单的相关性衡量；
   - 可扩展支持词频、TF-IDF 权重、BM25 等排序策略；
4. 性能特点：
   - 倒排索引查询时间复杂度低（O(1) 查词 → O(N) 合并列表）；
   - 空间利用率高，适用于大规模文本集合的检索任务。

本实验通过 Python 实现了一个轻量化倒排索引系统，支持词项存储、交并集查询、结果高亮与分页显示。

4. 中文分词原理

中文文本不同于英文，其词与词之间没有空格，需要通过分词算法将连续的字序列切分为具有语义的词项。常见的分词方法包括：

1. 基于词典的最大匹配法（如正向最大匹配 FMM）；
2. 基于统计模型的分词（如 HMM、CRF）；
3. 基于深度学习的分词方法（如 BiLSTM-CRF）；

本实验采用开源的 jieba 分词工具，其核心原理为：

1. Trie 树匹配 + 词频概率估计：将用户词典构建为前缀树，进行最大概率路径切分；

支持多种模式：

2. 精确模式（jieba.cut）：适用于构建倒排索引；
3. 全模式：用于关键词推荐；
4. 搜索引擎模式：保留短词，适合召回；

通过 jieba.cut(title) 处理每本书的标题，可有效提取关键词并用于索引构建，使搜索引擎具备良好的中文理解能力。

四、实验内容

1.网页爬取

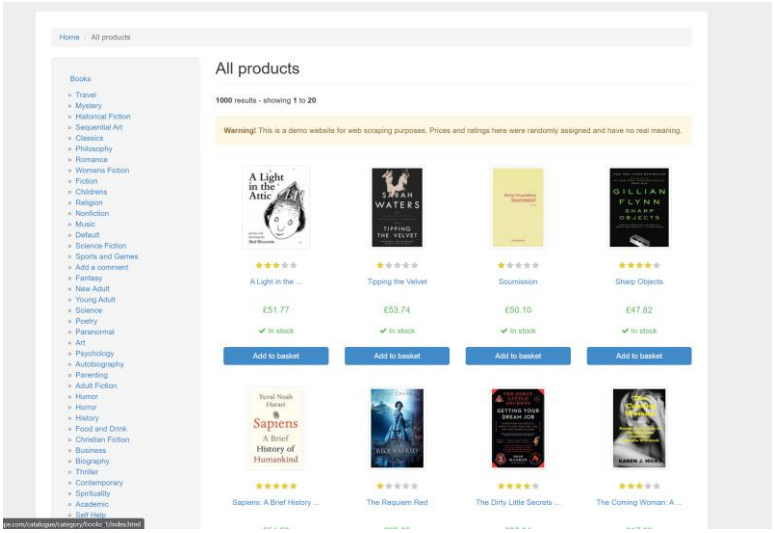
使用库：requests、BeautifulSoup、time、random、urllib

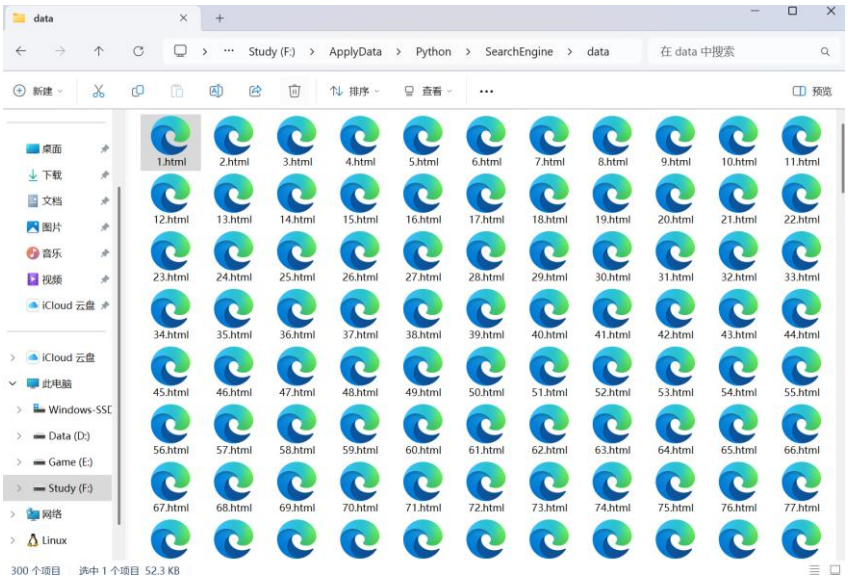
实现方式：

1. 使用 requests.get() 模拟浏览器请求网页；
2. 通过 BeautifulSoup 解析 HTML 页面结构，提取 <a href> 标签中的超链接；
3. 使用 urllib.parse.urljoin() 将相对链接转换为绝对路径；
4. 利用队列（BFS）遍历所有可达页面，同时使用集合 visited 去重；
5. 设置 User-Agent 避免被识别为爬虫，增加 sleep 实现请求间隔控制；
6. 抓取的网页 HTML 内容以 .html 格式保存在 data/ 目录。

作用：

该阶段实现对目标站点（books.toscrape.com）的完整抓取，为后续内容提取和搜索系统构建提供原始网页数据。





2.网页处理

使用库：BeautifulSoup、json、glob、os

实现方式：

1. 对 data/*.html 文件进行逐页解析；
2. 使用 BeautifulSoup 定位每本图书的 HTML 结构块 <article class="product_pod">；
3. 从中提取图书标题（title）、价格（price）、库存状态（availability）、评分（rating）和链接（url）等字段；
4. 所有结构化图书信息统一保存为 JSONL 格式，写入 books_extracted.jsonl 文件，每行为一条图书记录。

作用：

该阶段将半结构化的网页数据转换为标准化的结构化语料，便于后续翻译、索引和查询处理。

3. 构建中文图书语料库

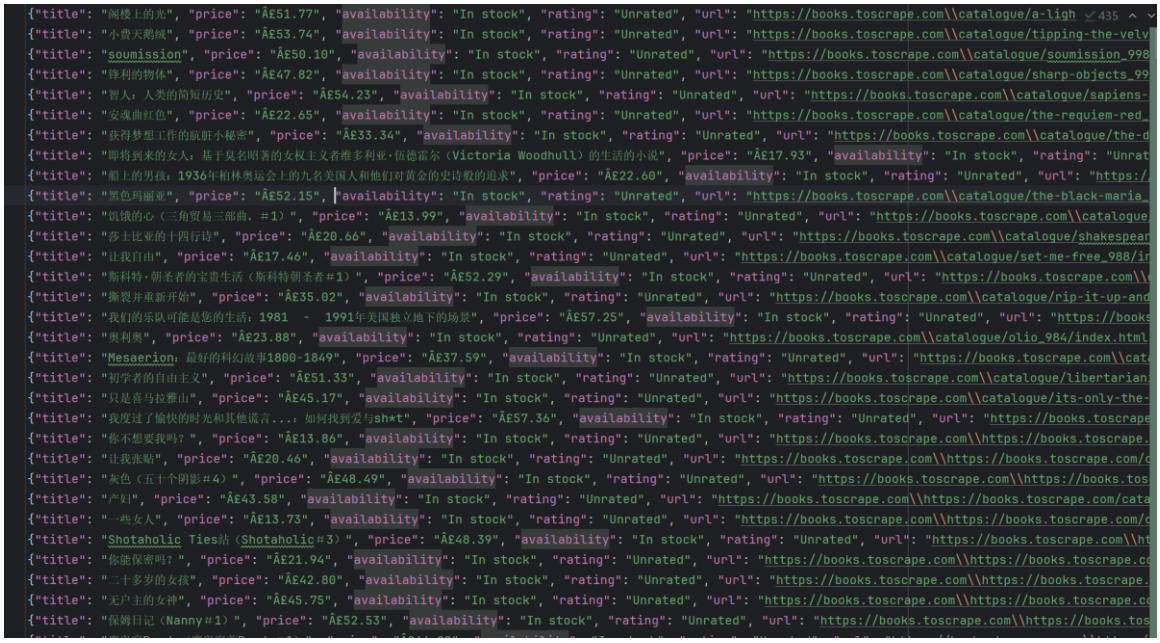
使用库：googletrans、json

实现方式：

1. 遍历 books_extracted.jsonl 中的记录；
2. 使用 googletrans 库调用 Google 翻译 API，将每条图书的 title 字段从英文翻译为中文；
3. 其余字段如价格、库存、评分、链接保持不变；
4. 最终结果保存为 books_zh.jsonl，结构与原文件一致，仅 title 字段为中文文本。

作用：

该阶段将英文语料转换为中文图书语料，为后续中文分词与检索系统搭建提供基础数据。



4.中文分词与倒排索引构建

使用库：jieba、json、collections.defaultdict

实现方式：

1. 对 books_zh.jsonl 中的每条记录的 title 字段使用 jieba.cut() 进行中文分词；
2. 使用 defaultdict(set) 构建倒排索引结构，将每个词项映射到包含它的图书记录编号列表；
3. 原始记录使用 doc_store 字典存储（doc_id → 图书记录）；
4. 所有倒排索引与文档内容统一保存为 inverted_index_zh.json，结构如下

```
{
  "index": {"红楼梦": ["0", "12"], "刘慈欣": ["1", "7"]},
  "docs": {"0": {...}, "1": {...}, ...}
}
```

作用：

倒排索引是检索模块的核心结构，使得关键词查询操作可以在常数时间内快速完成文档查找，提高搜索性能与系统响应效率。

```
{
  "index": {
    "阁楼": [
      "1622",
      "1626",
      "1619",
      "1103",
      "0",
      "1635"
    ],
    "小费": [
      "1625",
      "1657",
      "1597",
      "1629",
      "1618",
      "1621",
      "1634",
      "1"
    ],
```


5.中文检索模块实现

使用库：jieba、termcolor（用于高亮）、json

实现方式：

1. 用户输入中文查询语句，通过 jieba 分词解析为关键词列表；
2. 根据查询逻辑选择：
    - AND 查询：取多个关键词 posting list 的交集；
    - OR 查询：取 posting list 的并集；
3. 相关性排序方式：根据命中词条数进行降序排列；
4. 在检索结果中将命中的关键词进行红色加粗高亮（termcolor.colored）；
5. 分页输出，每页最多 5 条记录，支持用户输入 n 查看下一页或 q 退出查询。

作用：

该模块作为整个搜索引擎的“前台接口”，实现了从用户输入到结果展示的完整闭环，是用户体验最直观的部分，体现搜索引擎对中文语义理解与快速响应能力。

五、实验总结

本次《智能搜索引擎技术》实验以构建一个功能完整的中文图书搜索引擎系统为目标，涵盖了搜索引擎从数据采集、内容处理、语料构建、索引建立到检索交互的关键模块。实验初期，我们基于 Python 使用 requests 和 BeautifulSoup 自主实现了网页爬虫程序，采用广度优先搜索（BFS）策略进行 URL 调度，抓取了 books.toscrape.com 网站中的大量图书信息页面，并通过解析 HTML DOM 树结构提取出图书标题、价格、评分、库存状态等字段，形成标准化 JSONL 格式的中间数据集。

在此基础上，我们进一步使用 googletrans 将图书英文标题翻译为中文，构建适用于中文环境的检索语料库。为了处理中文文本的分词问题，我们引入了 jieba 分词工具，采用精确模式对标题内容进行分词切分，并构建了词项到文档编号之间的倒排索引结构，极大提高了检索效率。随后，我们基于倒排表实现了支持 AND/OR 查询逻辑的关键词检索系统，通过词命中数量进行相关性评分排序，同时引入 termcolor 实现检索结果的高亮显示，并支持分页交互输出。

整个实验不仅实现了搜索引擎基本功能的端到端流程，还在工程能力、模块协作、数据标准化处理等方面给予了我们深刻的训练。通过实验，我们不仅掌握了搜索引擎系统的技术路径和实现要点，也意识到中文信息检索中分词准确性、索引结构设计与查询逻辑控制等方面的重要性。本次实验极大增强了我们对信息检索、自然语言处理及系统开发实践的认识，为后续深入研究搜索引擎、语义理解与智能问答系统等方向奠定了扎实基础。

六、附录

详细代码详见我的 github：https://github.com/brainhuahua/Smart-search-engine

可以通过 https://github.com/brainhuahua/Smart-search-engine.git 克隆。