# 校园搜索引擎实验报告

2016011388 潘庆霖 计61

2016011398 高鸿鹏 计61

## 所使用的工具和代码框架

1. 网页抓取工具Heritrix1.14.4
2. Lucene 8.1.1
3. 分词工具： Lucene内置StandardAnalyzer
4. Html解析：Jsoup 1.21.1
5. PDF 解析：pdfbox 2.0.15
6. Doc 解析：poi 4.1.0
7. 前端服务：apache+tomcat (http://tomcat.apache.org/)

## 实验要求

- 抓取清华校内绝大部分网页资源以及大部分在线万维网文本资源（含M.S. office文档、pdf文档等，约20-30万个文件）
- 实现基于概率模型的内容排序算法；
- 文本检索实验已经让大家实现基于查询分词的VSM或BM25模型
- 建议改写提供的图片搜索框架或查找开源资源在其之上进行加工。
- 实现基于HTML结构的分域权重计算，并应用到搜索结果排序中；
- 实现基于PageRank的链接结构分析功能，并应用到搜索结果排序中；
- 采用便于用户信息交互的Web界面。

## 基本功能实现流程

### 1.使用Heritrix来抓取校园的网页

教程参考：

https://www.ibm.com/developerworks/cn/opensource/os-cn-heritrix/

在完成基础配置之后，将Heritrix改造成多线程抓取(50个线程)，参考教程如下

https://blog.csdn.net/yangding_/article/details/41122977

然后在网页端开启Heritrix服务器开始抓取，相关选项的具体设置如下：

```
Heritrix组件配置参考
Select Crawl Scope
org.archive.crawler.scope.BroadScope
Select URI Frontier
org.archive.crawler.frontier.BdbFrontier
Select Pre Processors
org.archive.crawler.prefetch.Preselector
```

```
org.archive.crawler.prefetch.PreconditionEnforcer
Select Fetchers
org.archive.crawler.fetcher.FetchDNS
org.archive.crawler.fetcher.FetchHTTP
Select Extractors
org.archive.crawler.extractor.ExtractorHTTP
org.archive.crawler.extractor.ExtractorHTML
Select Writers
org.archive.crawler.writer.MirrorWriterProcessor
Select Post Processors
org.archive.crawler.postprocessor.CrawlStateUpdater
org.archive.crawler.postprocessor.LinksScoper
org.archive.crawler.postprocessor.FrontierScheduler
```

设置抓取的种子列表为

```
http://news.tsinghua.edu.cn
```

设置过滤器不抓取图书馆资源：

```
[\S]*lib.tsinghua.edu.cn[\S]*；
[\S]*166.111.120.[\S]*
```

设置正则表达式的过滤项

```
.*(?i)\.(mso|tar|txt|asx|asf|bz2|mpe?g|MPE?G| tiff?
|gif|GIF|png|PNG|ico|ICO|css|sit|eps|wmf|zip|pptx?|xlsx?|gz|rpm|tgz|mov|MOV|exe|jpe?g|JPE?
G|bmp|BMP|rar|RAR|jar|JAR|ZIP|zip|gz|GZ|wma|WMA|rm|RM|rmvb|RMVB|avi|AVI|swf|SWF|mp3|MP3|wmv
|WMV|ps|PS)$
```

## 2. 文本网页数据，数据处理与PageRank的计算

所有文件见文件夹 `Search\searcher\src\url_measure`，各文件的用途如下：

- clean_url.py

  输入文件为crawl.log（Heretrix自带）,输出文件为cleaned_url.log,筛选需要的文件格式类型，过滤错误形式的网页

- get_all_information.py

  主要完成对htm/html结尾的文件的内容提取，将提取出的信息保存在anchor.log与title.log中，然后生成计算PageRank的网页关联信息，保存在tsinghua.graph中

- get_page_rank.py

  完成对PageRank的计算，主要依靠tsinghua.graph文件，首先计算出入度，再根据算法计算每个结点的PageRank信息，再读入get_all_information生成的题目和锚文本信息(仅html文件处理)，输出为pagerank.py文件

  排名前十的页面信息如下：

```
/publish/thunews/index.html 0.054811795153016936    首页  清华大学新闻网
/publish/thunews/9652/index.html    0.0443036459792194  更多 &#8250;  清华大学新闻网 - 图说清华
/publish/thunewsen/index.html   0.03675168257909178 ENGLISH Tsinghua University News
/publish/thunews/9650/index.html    0.026648308726768863    媒体清华    清华大学新闻网 - 媒体清华
/publish/thunews/10303/index.html   0.02524876605085597 综合新闻    清华大学新闻网 - 综合新闻
/publish/thunews/9649/index.html    0.02453537834353733 要闻聚焦    清华大学新闻网 - 要闻聚焦
/publish/thunews/9657/index.html    0.024534809277958085    新闻合集    清华大学新闻网 - 新闻合集
/publish/thunews/9656/index.html    0.024527182247498405    清华人物    清华大学新闻网 - 清华人物
/publish/thunews/10304/index.html   0.024485341293384307    新闻排行    清华大学新闻网 - 新闻排行
/publish/thunews/10237/index.html   0.023906930952428785    rss 清华大学新闻网 - rss
```

# 3. 构建检索及倒排索引

## 3.1 文档解析

### 3.1.1 HTML文件解析

实验中使用Jsoup工具包解析网页，抽取title标签的文本内容作为文档的的标题域；抽取p、span、td、div、li、a标签的文本内容作为文档的内容域；a标签的内容表示页面链出的内容，也作为一个anchorOut域单独索引；h1-h6标签的文本内容表示页面内的小标题，拿出来作为一个域；此外，进入页面的链接有着和页面标题相似的作用，单独成为一个anchorIn域。

### 3.1.2 PDF文件解析

实验中使用pdfbox解析文件获得内容域，直接以文件名作为标题域。(这也导致了搜索出来的pdf文件的标题有时候表现为一串没有规律的数字)

### 3.1.3 DOC文件解析

实验中使用POI包解析文件获得内容域，也是直接以文件名作为标题域。比较麻烦的是，POI工具解析.doc文件.docx文件的方法并不一样，因此需要在解析之前进行更细致的分类。

# 4. 检索

- 借鉴学长的经验，使用MultiFieldQueryParser将多个Field组合在一起进行查询。同时，由于Lucene4.0的版本有点老旧，我们采用了比较新的Lucene8.1.1,在API的使用上和图片搜索框架出现了一些不一致的地方。IK Analyzer由于太久没有更新，也无法使用了，我们将分词器改为了Lucene提供的StandardAnalyzer。
- 评分方法采用了Lucene中实现好的BM25评分。
- 通过人为观察效果，来确定各个域在搜索过程中的权重。最终将分域权重调整为100、25、35、1、0.1。（依次为标题域、h域、链入域、内容域、链出域）

# 运行方式

- 运行环境 : Ubuntu16.04

- IDE : IDEA
- jdk : 12.0.1
- 需要进行如下设置。
    - 用idea打开maven工程之后，由于不同的pc上tomcat的位置有可能不同，所以可能需要在idea中进行编辑。
    - 将pagerank文件放置在项目顶层目录下，将源代码中的MyIndex.java中的page_root改为爬虫爬取的文件所在的目录，MyServlet.java中的indexDir改为建立的索引的全局目录（与MyIndexer中main函数的给入参数一致）。
    - 运行配置好的项目，IDEA将会自动打开浏览器测试网页。