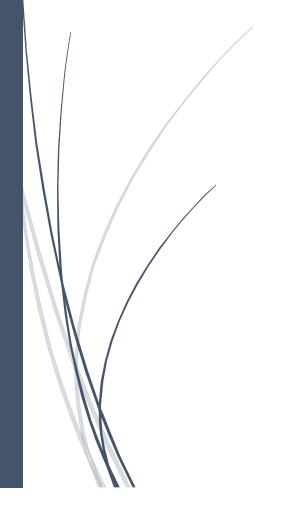
# 2014年6月4日

# 校园搜索引擎报告

《搜索引擎技术基础》课程



小组成员:

2011011237 张宏辉

2011011238 许建林

# 目录

<b>—</b> 、	实验目标	2
Ξ,	搜索数据处理	2
	. 1 官方网页爬取	
	. 2 人人公共主页爬取	
	. 3 文档解析与文档索引	
三、	搜索引擎后端	5
	. 1 BM25 多关键词多域查询框架	
3	. 2 GBRT 训练查询参数	5
	. 3 特色功能实现	
四、	搜索引擎前端	7
	.1 前后端完全分离	
4	. 2 前端功能实现	7
五、	搜索引擎效果展示	8
	. 1 欢迎界面	
	. 2 搜索关键字补全	
5	. 3 基本搜索结果	9
5	. 4 垂直搜索结果	10
5	.5 指定域搜索	11
六、	实验总结	12
七、	文件说明	12

### 一、实验目标

以清华各类官方网站与相关人人公共主页数据为搜索源,实现清华校园搜索引擎, 使搜索效果尽可能理想。同时尽量完善搜索引擎功能,使搜索引擎功能更全面更友好。



### 二、搜索数据处理

#### 2.1 官方网页爬取

本次实验需要抓取清华校内不包括图书馆的网页资源,采用 heritrix-1.14.4 进行抓取。抓取的种子:

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

从 student 出发,可以扩展出清华几乎所有的网站(509 个),最终抓取了近 76万个文档,总共 67GB,但由于很多都是 php,jsp等界面,而 Heritrix 在保存网页时命名方式和实际 URL(当有 get 参数时)不一样,所以即便把它们进行索引,同样无法获取有效 URL,所以这些页面我都没有进行索引,另外很多 pdf,doc 文档因为加密以及不符合规范,解析会失败,所以我把这些文档删除了(删除的部分并未包括在 76万个文档中)。

#### 2.2 人人公共主页爬取

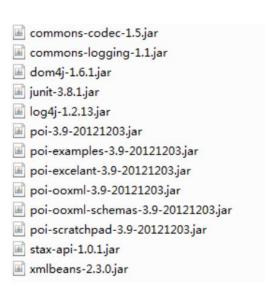
我们搜集了和清华的学习生活相关的 80 个人人公共主页(请见 data/renren. txt), 利用人人网开放的 API 获取这些公共主页的状态和日志,把这些数据进行了索引,补充了官方网站的数据源不足。最终我们获取了 8000 多条状态和 5000 多篇日志。

值得指出的是,在获取公共主页的日志时,由于人人网服务器的问题,总是发生Internal Error,所以我们采取了累进式抓取,失败后接着抓取。



#### 2.3 文档解析与文档索引

doc 解析使用 Apache POI, 使用包如下. 由于 doc 与 docx 格式不同, 所以需要分开进行解析:



pdf 解析使用的 pdfbox-0.7.3。使用包如下. 其中 pdfbox-0.7.3. jar 主要用于解析 pdf, 其他包与 pdf 中文编码乱码问题相关。

bcmail-jdk14-132.jar
bcprov-jdk14-132.jar
checkstyle-all-4.2.jar
FontBox-0.1.0-dev.jar
lucene-core-2.0.0.jar
pdfbox-0.7.3.jar

我利用了 jsoup 解析图片所在原始网页,加入到索引内容中。之所以使用 Jsoup,这是因为它一直在维护中,并且 Jsoup 支持类似 JQuery 的 CSS 选择器语法获取 DOM 对象,使用方便,而且之前已经用过很多次了,比较熟了。

这里我从每个 html 文件的内容中选择了5个域加入到索引内容中,分别是:

title, 即 html 的<title>标签的文本;

content, 文档内容, 由于 html 的复杂性, 实现很难达到满意的效果。实验中, 我将所有 p, span, td, th, div, li, a, pre, code, em, strong, b, i 等可见标签的文本组合起来作为内容:

subtitle, 即 h1~h6。由于几乎所有网页只包含 h1~h6 的一个, 因此我没有把它们拆开;

anchor\_out, 网页中包含所有超链接的文本。

anchor in, 即指向该网页的锚文本。

这里需要特别指出的是,网页文件的编码方式问题,我是采取了这样的解决方案: 因为基本上所有的网页文件都会在其 head 标签中指出编码方式,而这部分内容是英文的,所以可以首先采用 utf-8 编码解析出这部分内容,然后利用字符串查找,找到编码方式,然后再以正确的编码方式对该文件进行解析。当然,如果这个查找过程失败,就丢弃这个网页,因为基本上所有的网页都会指出其编码方式,所以这部分是可以忽略的。处理过程中发现只有极少数网页这一步骤出了问题,都是什么 gb2313, utf\_8, GB, gbk2312, gb2132, gb 2312-80 等低级的问题,由此可见互联网的世界是多么肮脏黑暗!

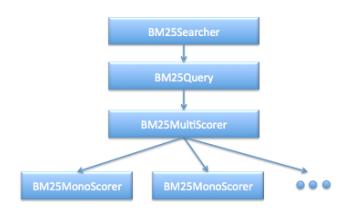
解析完成之后直接使用 Lucene 的索引工具即可, 建立完索引之后就是进行查询了。

### 三、搜索引擎后端

#### 3.1 BM25 多关键词多域查询框架

图片搜索实验中, 我们在 Lucene 框架下实现了简单的 BM25 查询框架。在此基础上我们重写了评分框架, 支持 BM25 的多关键字多余查询。

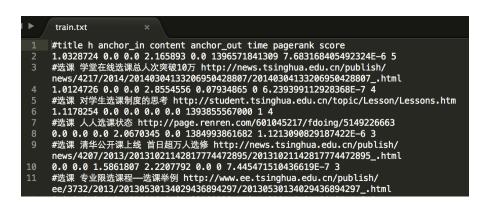
查询流程图:



上述流程图中每个图块都对应 src/search 下面的一个文件。BM25Searcher 是接口程序:构造函数载入索引文件,loadGlobals()载入全局参量,BMSearch()为搜索的接口函数。当搜索时会调用 BM25Query 获得 BM25Weight, 其中的 score()会生成 Scorer,用于生成所有符合条件的文档并计算分值。score()函数对分词后的每个 term 先调用单关键词查询 BM25MonoScorer,然后进行多关键组合 BM25MultiScorer,这里的多关键字指目标文档必须同时包含这多个关键字。这两个 Scorer 的实现与图片搜索类似,应用到的机器学习方法将在下文 3. 2 中说明,具体可以参考实现代码。

#### 3.2 GBRT 训练查询参数

BM25MonoScorer 的评分结果是搜索引擎搜索效果的重点。评分时我们需要涉及五个域的 BM25 Score (包括 title, h, anchor\_in, content, anchor\_out), 此外还需要涉及 PageRank, Time 两大因素。我们选择用 GBRT (Gradient Boost Regression Tree)来学习这7个参量之间的参数。



GBRT 的源代码可以在网上下载,放置在 src/gbt 里,但我们需要足够的训练集来训练模型。训练集见 data/train. txt 文件,我们手工标注了 110 条搜索结果,用 1-5 进行了打分。GBRT 的训练与模型的载入与导出都在 BM25MonoSearcher 里面实现,导出的训练结果文件见 data/ScoreGbrt. txt.

#### 3.3 特色功能实现

为了满足前端特色功能的实现,后端在 BM25Searcher 中实现了 genAbstract()获得摘要, getRelated()获得历史记录中的相关搜索(Json 格式), BM25Search()获得搜索结果(Json 格式,包含一系列特殊信息)。

关于 Json 格式返回信息做如下说明:

#### # BM25Searcher()

▶ result: 搜索结果,包括 num, title, text, url, unique

➤ related: 相关搜索

▶ picSpecial:图片型垂直搜索结果,包括text, url, pic

▶ textSpecial:文字型垂直搜索结果,包括title, content

#### getRelated()

> related: 相关历史搜索记录

#### getTop()

> result: 返回热门搜索词

### 四、搜索引擎前端

#### 4.1 前后端完全分离

为了减少合作开发过程中的耦合,我们采用了前后端完全分离的架构,后端通过 http server 向前端提供信息,全部采用 get 请求的方式。而前端则是 Python Django 框架提供一个 web server,在用户浏览器端,采用 JQuery 向这个 web server 发起查询。 实际上这个 web server 只是一个服务器端的代理,它将和后端进行数据交换,转发用 户的 Query 以及后端返回的结果。

这样设计的好处不但减小了开发过程中的耦合,而且使得前后端可以分别部署到不同的服务器,而后端则可以提供 REST API 的形式为其他开发者提供一个 web service。

#### 4.2 前端功能实现

前端的实现主要采用 JQuery 库,通过 js 脚本来实现数据通信和数据跳转,使得整个搜索界面都在一个网页内,这样做可以大大简化用户 cookie 等数据的传递开销,使得进一步的开发更加容易。而 UI 的设计则是采用了流行的 bootstrap 框架,使得 UI 更加友好,而不是只有一个框,一个钮,其余部分全是空白。

垂直搜索是通过后端给出的 API 来实现的,每次查询的时候,后端返回的结果都可能包含一个垂直搜索结果,通过特判就可以获知是否有垂直搜索结果,有的话就进行特殊展示。

搜索关键字补全,后端会提供一个相关搜索的 API, 返回与当前用户输入的字符串前缀匹配的 Query, 然后前端会在搜索框下面对用户进行提示, 至于发送这个 API 请求的时机,则是采用了目前很成熟的方式,通过回调函数,当用户输入暂停一定时间之后,在这个回调函数内发送 API 请求,这样可以减轻服务器压力,但是并不影响用户体验。

页面去重处理,后端返回的结果中,会包含一个标签,显示这个结果是否重复,如果重复则先将其div设置为不可见,当用户点击继续查看后将这些div设置为可见即可。

指定域搜索则是后端在处理 Query 时,对 type:xxx 开头的 Query 会进行类型过滤,所以当用户指定搜索类型之后,只需要在其输入的关键词之前加上相应的 type 即可。

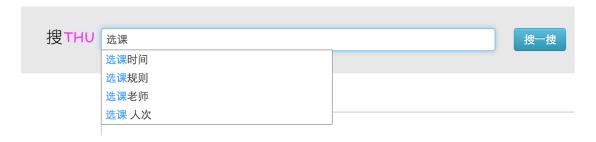
页面上所有的关键字高亮都是在前端通过匹配结果的摘要及标题中的查询词,然后调节 css 样式来实现的。

# 五、搜索引擎效果展示

### 5.1 欢迎界面



## 5.2 搜索关键字补全



#### 5.3 基本搜索结果

搜索"校庆"(搜索效果理想,时间符合,页面去重实现):



#### 已经为您隐藏部分相似度极高的结果,如需要继续查看,请点击这里

搜索"PX"(搜索效果理想,人人信息补充官网信息):



#### 5.4 垂直搜索结果

搜索"放假"(出现清华校历供查询放假时间):



搜索"搜索引擎上课"(出现搜索引擎基础这门课程的基本信息):

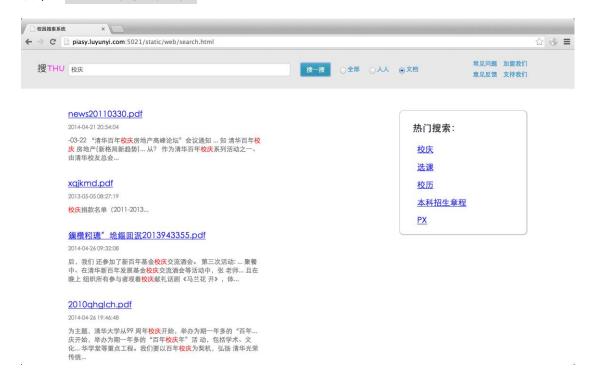


#### 5.5 指定域搜索

#### 搜索"校庆(指定人人域)":



#### 搜索"校庆(指定文档域)":



# 六、实验总结

总结本次实验, 校园搜索引擎主要亮点为如下三点:

#### ▲ 索引数据量丰富

- ✓ 索引 26 万文档,索引文件 2.02G
- ✔ 来源官方网站、人人公共主页等

#### ▲ 搜索效果理想

- ✓ 考虑时间因素
- ✓ GBRT 学习参数

#### ▲ 友好功能多

- ✓ 垂直搜索返回
- ✓ 关键字补全,页面去重
- ✓ 热门搜索,相关搜索,指定域搜索

通过本次实验,我们对搜索引擎构建的步骤有了更清楚的认识。再次感谢老师与助教在实验中给予的指导与建议!

# 七、文件说明

文件内容如下:

文件夹	文件(夹)	说明
doc/	校园搜索引擎报告. pdf	实验报告
	slides.pptx	展示 PPT
	校园搜索实验说明-2014. pdf	实验说明
THUSearch/	run. sh	Linux 下运行脚本
	THUSearch. jar	部署文件
	frontend/	前端代码
i nosear cn/	src/	后端代码
	index/	索引文件
	data/	包括 GBRT 训练集等