校园搜索引擎构建 设计文档

计45 李昊阳 2014011421
计45 王龙涛 2014011406

1. 实验介绍

1.1 实验题目

校园搜索引擎构建

1.2 实验内容

综合运用搜索引擎体系结构和核心算法方面的知识，基于开源资源搭建搜索引擎。

1.3 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；
- 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；
- 尽可能尝试扩展功能。

1.4 实验环境

- Eclipse内置Tomcat8
- Ubuntu 16.04.

2. 项目介绍

2.1 项目准备

需要安装maven工具，以下载依赖包

```
$ mvn clean
$ mvn package
$ mvn dependency:copy-dependencies
```

2.2 项目配置

- Tomcat8在*Eclipse*中的配置,流程如下

1. 创建Runtime Environment

```
1. Window -> Preferences -> Server -> Runtime Environments -> Add...
    -> Apache Tomcat v8.0 -> Next
2. 指定Tomcat8.0的安装根目录和Java JRE的安装路径，点击完成即可。
```

2. 创建并配置Server

```
1. Window -> Show View -> Others... -> Server -> Servers -> OK
    打开服务器窗口
2. 右键 -> New -> Server -> Tomcat v8.0 Server (Server 's host name,
    Server name取默认即可, Server Runtime 选择上一步创建好的Tomcat v8.0运行环境)
    -> Next -> 将本项目工程添加至右方 -> Finish
3. 在左侧项目导航栏中可以看见有个Servers项目工程，右键 -> Run As -> Run Configurations...

    Arguments -> Working Directory: -> 将工作目录改为项目根目录
```

- Eclipse 配置项目

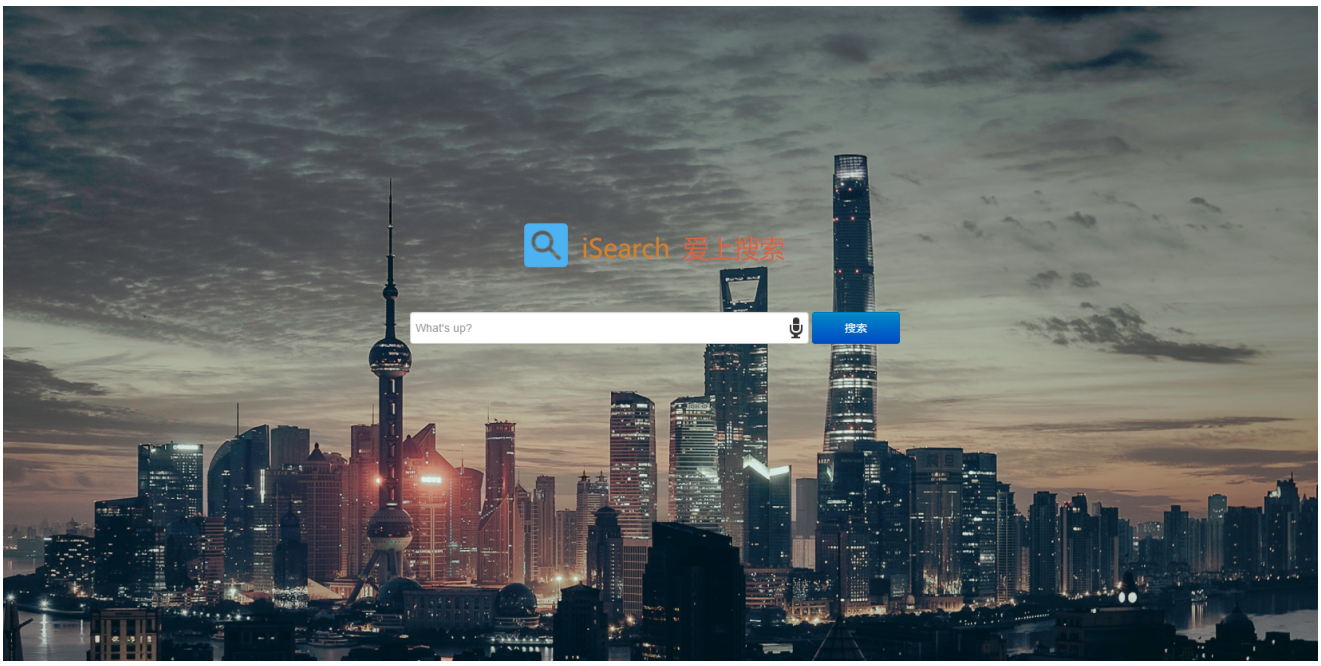
```
1. 打开项目 -> Dynamic Web Project  -> 选择Target Runtime选择Tomcatv8.0 ->
    两次Next  -> 修改WebContent至WebRoot  -> Finish
2. 添加jar包，位于target/dependency 目录下
3. 配置部署文件夹(Deployment Assembly) :
    /build/classes           WEB-INF/classes
    /target/dependency       WEB-INF/lib
    /WebRoot                 /
```

2.3 项目结构

```
├── build/
│   └── classes/                java编译生成文件目录
├── conf/                       配置文件目录
├── forIndex/                   索引目录
├── pom.xml                     maven配置文件
├── README.md                   README
├── report.md                   项目报告
├── src                         项目源码
├── target/
│   └── dependency/             jar包所在位置
└── WebRoot/                    网站根目录
    ├── servlet/                网站css,js等静态文件目录
    ├── thusearch.jsp           搜索主页
    ├── thushow.jsp             搜索页面
    └── WEB-INF/                网站配置文件目录
```

2.4 项目展示

2.4.1 主页面



2.4.2 搜索结果显示页面

3. 实验工具

因为本项目使用maven下载依赖包，所以实验使用的工具均可以在pom.xml文件中进行查看

3.1 Heritrix

数据抓取工具

3.2 lucene-core

网站开源框架

3.3 WebCollector

网页爬虫框架，使用其中的正文提取功能

3.4 中文分词工具IKAnalyzer

中文分词工具

3.5 ansj_seg

中文分词库，效果相比较IKAnalyzer更优

3.6 pdfbox, bcprov-jdk15

pdf解析工具

3.7 poi, poi-ooxml-schemas, poi-scratchpad, poi-ooxml

微软文档(doc,docx)解析工具

3.8 jsoup

html网页解析工具

3.9 javax.servlet-api

java servlet所需的jar包,调试用

4. 基本功能

4.1 数据抓取

4.1.1 Heritrix

基本上同介绍ppt上面所说配置相同，把接收的url规则从`news.tsinghua.edu.cn`改为了`*.tsinghua.edu.cn`,并且种子也新增了如下：

```
http://news.tsinghua.edu.cn/  # 清华新闻
http://info.tsinghua.edu.cn/  # 信息门户
http://yz.tsinghua.edu.cn/    # 研究生招生网
http://life.tsinghua.edu.cn/  # 生命科学院
http://www.tsinghua.edu.cn/   # 官网主页
http://www.sem.tsinghua.edu.cn/  # 经管主页
http://www.law.tsinghua.edu.cn/  # 法学院主页
http://www.tup.tsinghua.edu.cn/  # 出版社
http://postinfo.tsinghua.edu.cn/node/  # 内网信息
http://academic.tsinghua.edu.cn/  # 教学门户
http://learn.tsinghua.edu.cn/    # 教学门户
http://friend.cic.tsinghua.edu.cn/  # 计算机实验室主页
http://student.tsinghua.edu.cn/  # 学生清华
http://myhome.tsinghua.edu.cn/  # 我们的家园
```

可能是种子太多的缘故，我们爬取的url速度很慢，目前总共爬取了31G,共计8万个文件，4万个html，经过测试发现搜索结果还算不错之后就停止继续爬取了。

4.2 基于概率模型的内容排序算法

图片检索实验中使用的`lucene`版本为3.5.0,版本过老，原先我们项目是基于图片检索实验框架的，但是后来因为maven支持的`IKAnalyzer`版本过高，同`LUCENE_35`不兼容，所以我们把`lucene`包升级至4.7.2。那么原先的实验框架需要大改，我们重新对实验框架进行调整，花费了较多的时间和精力。新框架使用`lucene47`内置*BM25*算法进行内容排序。

4.3 基于HTML结构的分域权重

建立索引的时候，我们对html结构进行了分析并不同的域，详细情况如下：

```
1. title : html的标题属性设置为title域
2. keywords : 对html中的h1-h6单独设置一个keyword域
3. content : 网页正文内容是个很难去抽取的工作，所以我们调用了一个库WebCollector，它是一个爬虫
             框架，但是其中有个正文抽取的功能效果很不错，报告上说有99%的正文抽取正确率，
             我们随机抽样了几个网页内容进行查看，发现效果确实不错。
4. links : 网页存在许多链接，链接上面的文字本身也是一种可以参考的信息，所以我们对于所有的
           `a`标签也建立一个links域
```

> 注意：对于pdf,doc等之类的文档来说，我们只建立了title和content域。

最后，对于各个域，通过小范围的数据测试结果，我们最终将权重设置如下：

```
<title : keywords : content : links>  = <100.0f : 10.0f : 5.0f : 1.0f>
```

能够得到一个比较好的搜索结果，使得标题符合搜索关键词的网页能够更加靠前，同时，关键词和内容匹配的更全面的网页也能取得一个比较好的评分。

4.4 基于PageRank的链接结构分析

我们在建立索引的时候，首先对于网页内容的链接结构进行分析，然后再调用pagerank接口离线计算各个网页的pagerank值。

PageRank的计算公式为

$$PageRank^{(k)}(n) = \alpha \times \frac{1}{N} + (1-\alpha) \times \sum_{i \to n} \frac{PageRank^{(k-1)}(i)}{Outdegree(i)}$$

本实验中我们设定参数$alpha = 0.15, N = 20$。

将计算出的*pr*值作为`lucene`的各个document的*boost*值，同lucene的*BM25*算法相结合起来，能够取得更优的排序结果。

下面是两个搜索结果的例子，可以看出无论是给出结果的完整性还是顺序性，搜索引擎的表现都比较不错。

从上图结果可以看出，搜索的时候，官网出现的概率变大了许多，那是因为官网存在许多入链，增加了pagerank评分，从而使的搜索结果变得更靠前的缘故。

5. 扩展功能

5.1 前端美化

- 使用Bootstrap美化前端界面；
- 使用Ajax增强实时交互的能力；
- 各个部件的布局参照Google，Baidu，搜索框和相关搜索部件悬浮于界面上不随滚轮滑动而滑动；
- 不同尺寸大小的窗口的适应；
- 搜索结果的正文或标题高亮；

5.2 多类型文档解析

我们实现了对不同类型文档的解析：doc, docx, pdf, xml等。

上图中搜索`饥荒`一词时，出现两个看似奇怪的网页，打开链接才知道，原来是两份pdf，这两份pdf中均有`饥荒`一词，下图是第二个链接打开之后的搜索`饥荒`的结果。

5.3 图片显示

考虑到搜索结果的图片可以给用户更为直观的感受，所以我们在搜索结果旁边显示了网页中的图片（如果网页中有图片的话）。当然，如何在网页的众多图片中选择最有代表性的图片是一个关键的问题。我们发现，一个网页中往往存在着许多没用的文字和图片，而这些文字、图片和网页的主要内容基本是没有关系的，比如对于一些边边角角上的图片等，它们不应该作为这个网页的代表图片。所以我们首先进行了正文提取的工作，即识别一个网页中哪些

部分是有用的*content*，哪些内容是没有用的，这里我们使用了`WebCollector.jar`这个包，它是一个用于网页内容爬取的jar包，我们使用了其中的正文提取部分的功能。

下面是搜索“师资队伍”的结果，可以看出几个关于清华教授个人信息的结果都给出了正确的图片：

5.4 分词改进

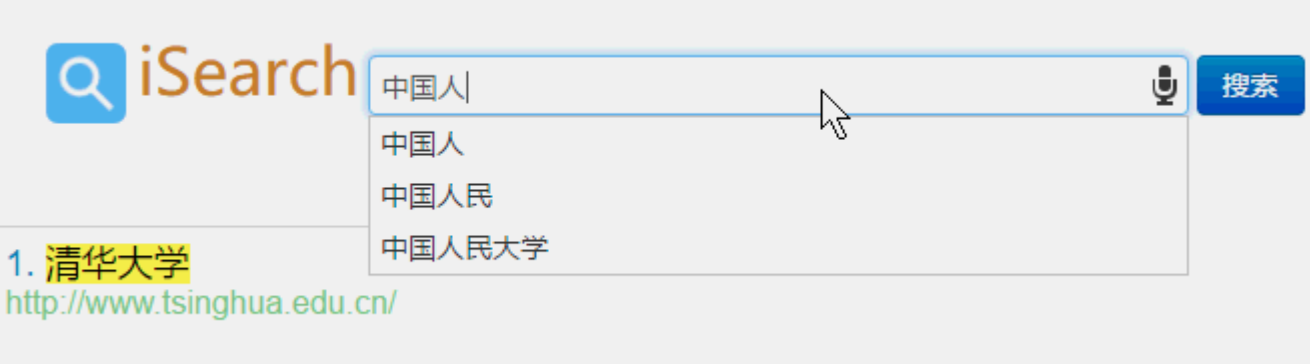
在原有的实验框架中使用的分词工具是`IkAnalyzer`，但是我们发现这个工具一些词汇上的分词结果并不好，所以我们就尝试使用了另外的中文分词工具`Ansj`。为了更好地对比这两个分词工具的性能，我们选择了若干查询词，对比使用这两种工具后分别给出的结果：

分词结果	IkAnalyzer	Ansj	哪个更好
贵系	<系><贵>	<贵><系>	一样
计算机	<计算机><计算><算机>	<计算机>	Ansj
方便面	<方便面><面><方便>	<方便面>	Ansj
王龙涛（人名）	<王><龙><涛>	<王龙涛>	Ansj
什么搜索引擎好	<什么><搜索><引擎><好><搜索引擎><索引>	<什么><搜索引擎><好>	Ansj

从上面的表格中可以看出，IkAnalyzer具有分词过于碎片化，存在重复的分词结果，而且不能识别人名的问题，但是Ansj在这些方面表现的都更好。改进了分词结果之后，搜索出的结果也会更能符合用户的要求。

5.5 查询词自动补全

查询词自动补全的功能是用户每输入一个字或者词，就搜索与当前查询词具有相同前缀的词汇并显示给用户。



前端部分时刻检测用户输入，当用户的查询词发生变化时就通过Ajax传给后端，后端进行检索并返回给前端`json`格式的列表，即为自动补全的词汇列表。

后端自动补全词检索算法如下：

1. 提取所有网页正文单词
2. 统计计算单词的词频
3. 按照词频进行排序，取出Top 6000的词汇，存入文件
4. 前端提供搜索词x，按照排序进行搜索前缀为x的单词列表返回

5.6 查询词纠错

该功能是搜索引擎常见的功能，用户的输入可能因为某些原因出错，搜索引擎需要在用户一部分查询词输入错误的情况下帮用户找到他可能要查询的词。

下面是一些纠错的结果：

后端进行查询词纠错使用基于Q-gram的到排列表算法，

首先计算两个字符串的距离,定义如下：

> 假设两个字符串能通过 k 次修改、添加、删除一个字符的操作互相转化, 则称这两个串的编辑距离为 k。

求串 a 和串 b 的距离可以使用动态规划的方法。用 $f[i,j]$ 表示串a 的前 i 个字符和串 b 的前 j 个字符之间的编辑距离。那么:

- a 进行添加操作(b 进行删除操作),转移到 $f[i+1,j]$;
- a 进行删除操作(b 进行添加操作),转移到 $f[i,j+1]$;
- a 进行修改操作(或不操作),转移到 $f[i+1,j+1]$ 。

最后 $f[|a|,|b|]$ 就是串 a 和串 b 的编辑距离。

考虑到对于每一个查询词，同所有串计算距离的效率太低。改进之后，先统计所有子串，两个相近的字符串，必定有很多相似的子串。因而取出串a的所有连续长度为Q的子串*Q-Gram*,如果a,b相似，那么b一定包含a的很多子串。

基于这个思路，我们建立了一个所有单词子串的倒排列表。当存在一个搜索词`query`的时候，拿串`query`的所有子串去倒排列表中搜索，取出和`query`至少拥有T个公共子串*Q-Gram*的字符串，那么这些串是很有可能与`query`相似的。接着在对这些串进行动态规划求出距离，最后将距离小于ED的词按照词频排序，取出前几个单词。

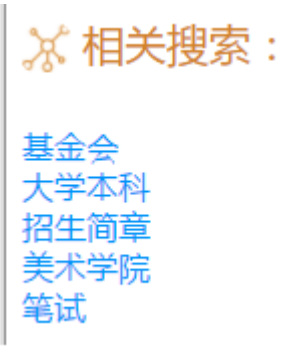
实际操作过程中,由于长的词需要纠错的字符个数大,短的词需要纠错的字符个数小,因此 ED 值会设为查询词长度*0.2 这样动态的值,以保证纠错的合理性。

5.7 相关词推荐

该功能同样是搜索引擎的常见功能，用户在输入一个词之后可能还会搜索和当前查询词相关的词，从方便用户的角度考虑，搜索引擎有必要给出一些推荐搜索的词汇。

下面是一些相关词推荐的结果：

搜索“清华”给出的相关词推荐：



搜索“图书”给出的相关词推荐：

搜索“金融”给出的相关词推荐：

首先定义相关性：如果两个单词出现在同一个网页中，那么他们被定义为是相关的。

假设查询词为 q,定义关键词 p 和查询词 q 同时出现的文档个数为$doc_{freq}(q,p)$。如果直接使用$doc_{freq}(q,p)$作为相关度的度量,那么会出现所有查询词的相关词都是“的”、“是”等没有信息量的词,因为他们几乎出现在所有的网页中。

这时候可以借鉴一下 TF/IDF 模型,将$doc_{freq}(q,p)$作为 TF 的值,p出现的网页次数记为$idf(p)$,通过求TF/IDF 的值来平衡常用词带来的干扰。但是这时又会出现一个新的问题,如果一个很生僻的词恰好和查询词在某一个网页中共同存在,那么它的TF/IDF 值就是 1。然而根据定义,TF/IDF 的值不会超过 1,因此这个生僻词就会成为最佳答案,然而其实它和查询词并不相关。

因此我们对 TF/IDF 公式做了一个修正,最终关键词 p 相对于查询词q 来讲其相关度定义为

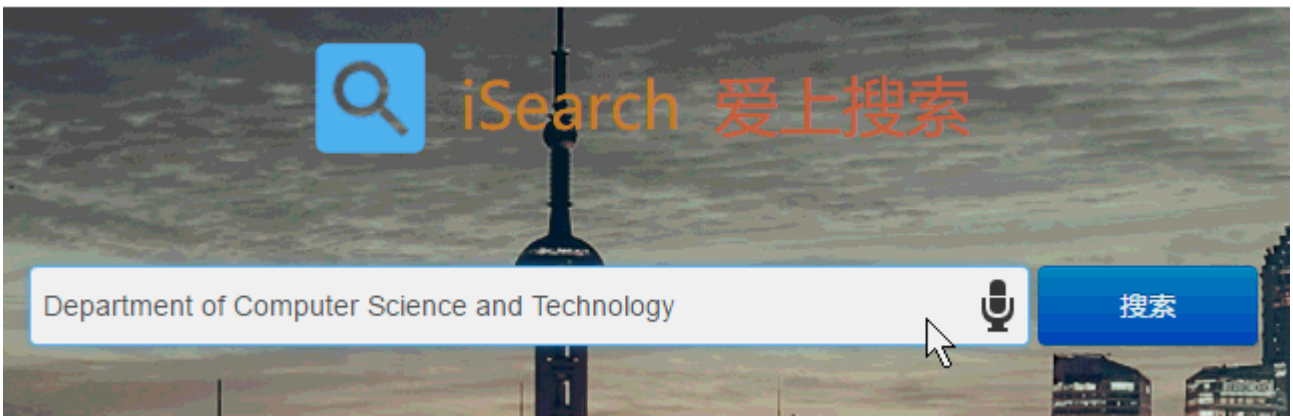
$$r(p,q)=\frac{doc_{freq}(p,q)}{\sqrt{idf(p)}}$$

这个公式中,即修正了常用词频繁出现的问题,也解决了生僻词被选为最佳答案的问题,再根据实际数据调整参数之后，总体效果显示不错。

5.8 语音输入

我们注意到Google提供了语音输入的功能，可以在用户输入时提供很大的便利，因此我们也实现了这个功能。用户可以点击搜索框右边的“话筒”按钮（如下图），开始使用语音输入的功能。用户说出要查询的关键词即可，如果用户2s中内没有说话则认为用户语音输入结束。

我们实现的语音输入除了基础的语音识别功能之外，还具有自动识别用户语言的功能（目前的默认设置的识别语言包括中文、英文）。此外，随着用户说出的词的不断增多，之前的词也会自动调整成更合适的选项，从下图可以看出语音输入可以较为准确地识别出用户说的查询词。



注：使用语音输入功能是需要联网并且用户打开麦克风的使用权限，如果是第一次点击录音按钮，浏览器会给出提示框询问是否允许使用麦克风，选择"允许"即可。

6. 实验感想

转眼之间，随着本次大作业的完成，《搜索引擎技术基础》马上就要结课了。回顾过去几周的开发过程，我们从会用搜索引擎到会写一个简单的搜索引擎，从对一些开源工具一无所知到能够熟练使用，从遇到错误就要调试半天到能够较快地定位的bug产生之处，可以说是让我们收获颇丰，受益匪浅的几周。我们不仅学习了数据抓取工具Heritrix，搜索引擎框架搭建工具Lucene，学习了Jsoup，pdfbox等解析工具，更对tomcat+jsp这一整套机制有了更全面理解，对搜索引擎背后的工作机理有了更深的认识。

刘老师上课的时候给我们深入浅出地讲解了很多搜索引擎的理论知识，而这次大作业就是将理论转化为实际最好的机会，通过一次次的实验，通过一步步的调试，我们不断加深着对理论知识的掌握程度，提高着自身的知识水平。

因为这门课是我们本科阶段最后一门限选课，这次大作业也是本科阶段最后一次大作业，所以我们都十分珍惜这次机会。虽然这次大作业没有软工项目那么复杂，没有计原造计算机那么艰难，但是它却依然给我们留下了非常深刻的印象，我们不会忘记这次富有挑战性的过程，不会忘记为了前端 的一点点优化就调到深夜，不会忘记完成大作业时的成就感和心中的喜悦，我们有理由相信这次大作业必定成为多年之后美好难忘的回忆。

最后向耐心为我们答疑解惑的老师和助教表示衷心的感谢！