Web信息处理与应用实验一程序说明——金泽文（PB15111604）

**简单的搜索引擎实现**

**【实验内容】**

通过Java语言,使用Lucene制作一个简单的搜索引擎,

其中创建索引以及查询的分词工具,使用中科院中文分词工具ICTCLAS,

再编写简单的jsp查询界面.

搜索引擎的数据库使用新浪网5个季度的新闻网页.

**【实验环境】**

编程语言: Java SE8, jsp

编程环境: Win10 Pro 1703, IntelliJ IDEA Ultimate 2017.3

运行环境: JRE 1.8.0\_152, JVM OpenJDK 64-Bit Server VM.

使用工具: Apache Lucene7.1.0,

NLPIR ICTCLAS 6.6.0,

Apache Tomcat 9.0.1

IntelliJ IDEA Ultimate 2017.3

Chrome 62.0

**【实验步骤及方法】**

主要步骤:

1. 调研与学习:

学习java.(这部分主要是通过 <Core Java>这两本书, 以及算法课的实验(故意用java写实验来学习并熟练))

调研当下业界认可的java IDE,以及业界认可的分词器.

调研并学习Lucene框架和ICTCLAS工具的配置,使用方法.(这部分主要是通过官方的github上的demo以及lucene官方文档进行学习)

调研并学习网页jsp的语法规则,以及Tomcat的配置与使用方法

1. 配置环境:

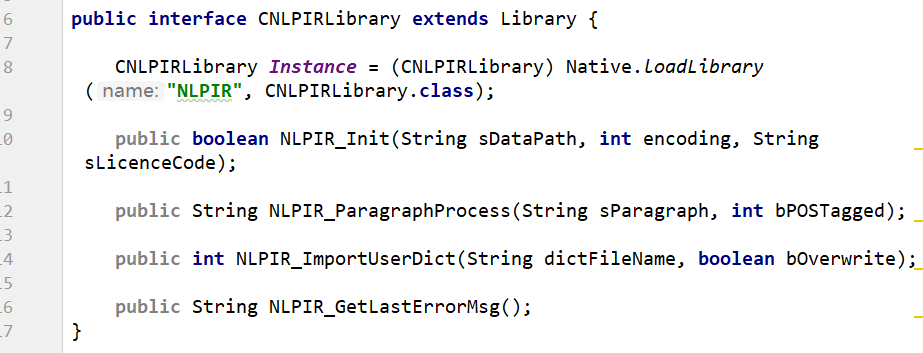
下载IntelliJ IDEA, 进行配置.

下载Lucene和ICTCLAS,以及tomcat,添加到环境变量中,并且配置项目的路径.

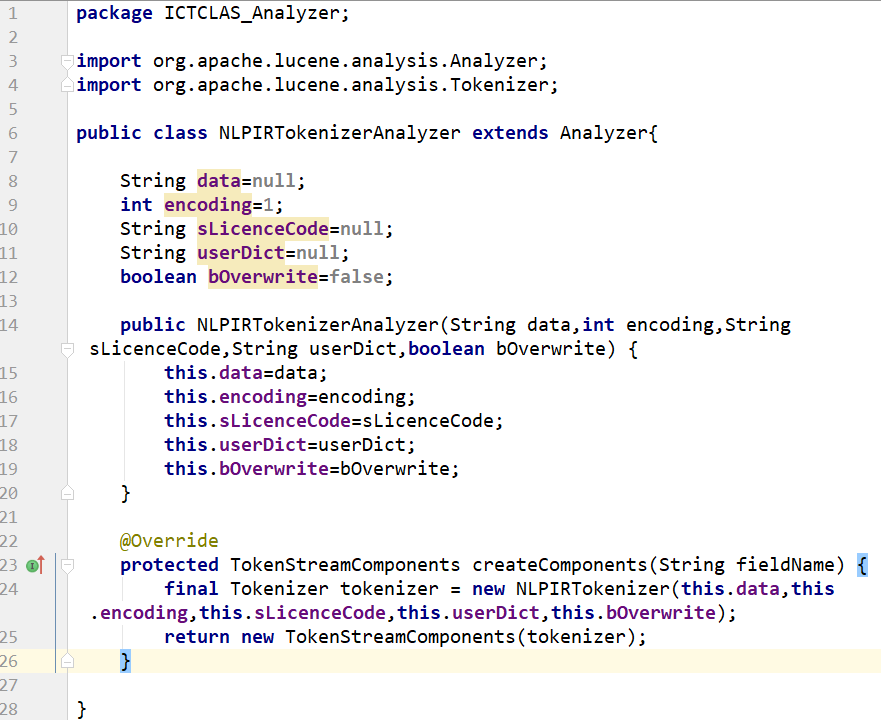
值得一提的是,中间因为IntelliJ和Tomcat的包路径等问题,调试了好几天.

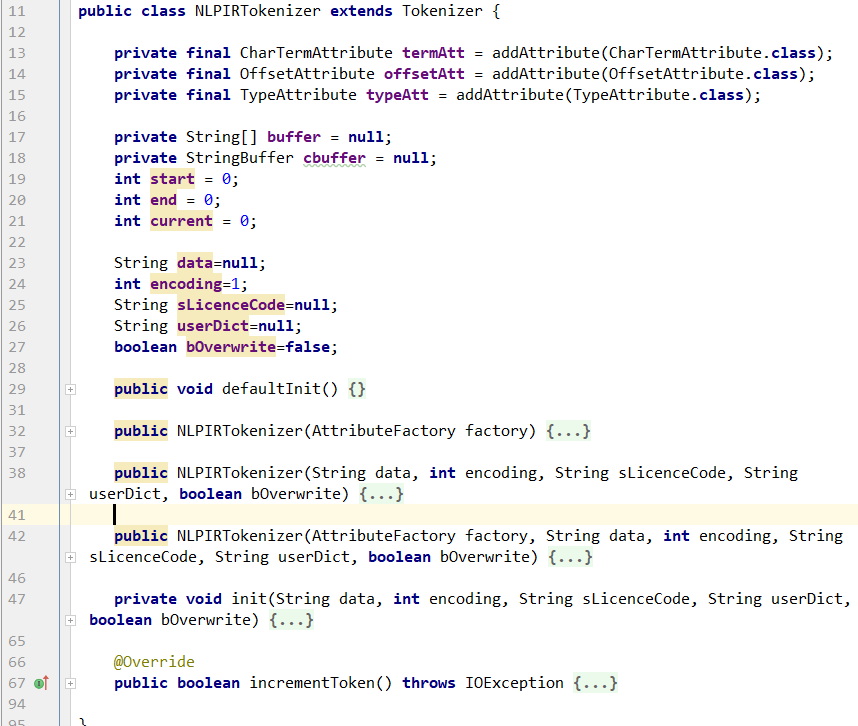
1. Coding和debugging
2. ICTCLAS的封装

接口类CNLPIRLibrary的封装



NLPIRTokenizerAnalyzer的封装

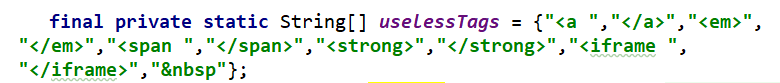


NLPIRTokenizer的封装与实现

1. 网页预处理的实现

·过滤无效标签:

对于每个文件的每个<doc>…</doc>部分进行处理.通过filterTags方法,过滤掉无效标签.

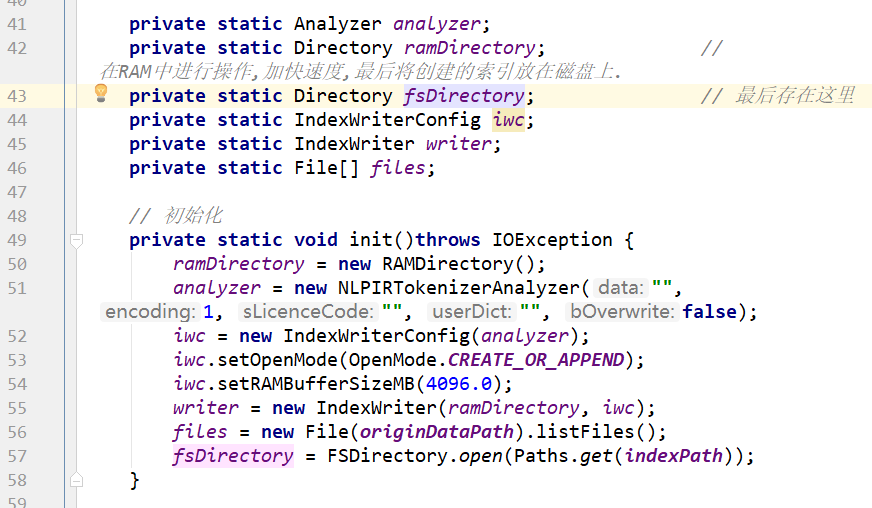
删除的标签存在数组中,其中包括5对标签和一个”&nbsp”,另外还有防止遗漏的正则表达式.

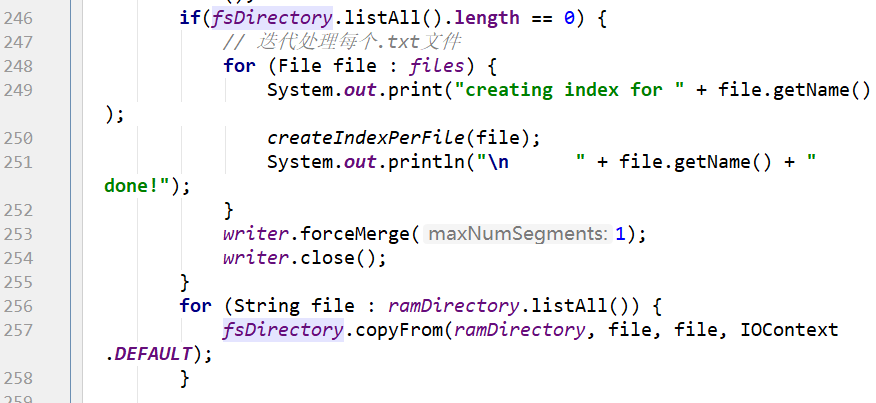
·提取有用标签

通过正则匹配与html解析工具Jsoup进行有用标签的提取,加入索引文档中.

3. 创建索引的实现

仿照官方demo,得到:

索引创建的中间过程在RAM中进行,最后再输出到磁盘系统中,提高性能.使用的analyzer为上面实现的NLPIRTokenizerAnalyzer.用analyzer设置IndexWriter,之后writer.add()的过程入上面选取有效标签之后加入的代码片段,不再赘述.

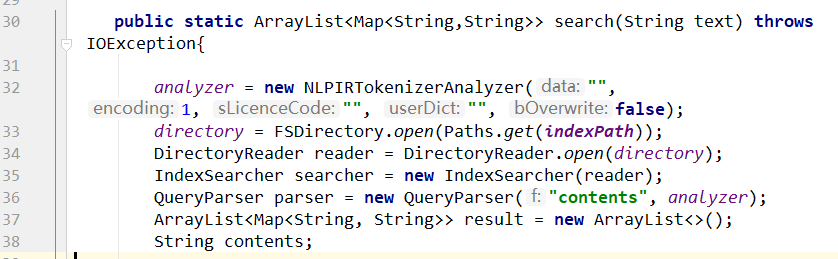
下面是main函数部分

迭代为每个文件创建索引之后,从内存中将索引结果转入磁盘中,以便后面搜索.

1. 查询的实现

·高亮的实现：我的高亮没有使用现成的highlight类，而是自己利用NLPIR的接口先进行分词，然后进行匹配.

·查询模块 - search函数的实现:

search函数返回ArrayList.下面是初始化:

然后是根据index得到的标签通过hitDoc.get(“title”)等方式得到对应字符串,并且通过map.put(”title”,str)的方式加入HashMap中,最后再加入到ArrayList中,并且返回.

1. 搜索界面的实现

由于我不是对前端设计很感兴趣,所以这部分没有花费太多精力设计.

代码如下:通过post方法,传递搜索串到result.jsp

1. 搜索结果界面的实现

这部分主要是嵌入的java代码.

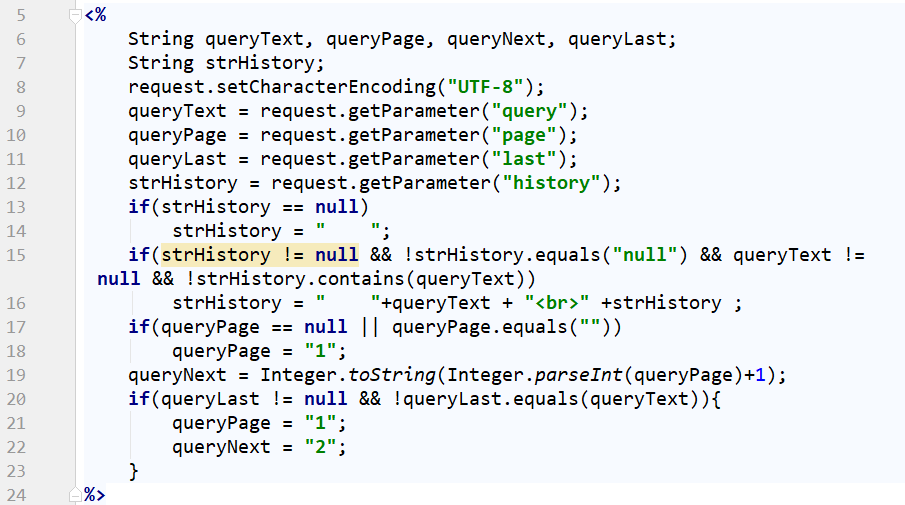
首先读取几个参数.

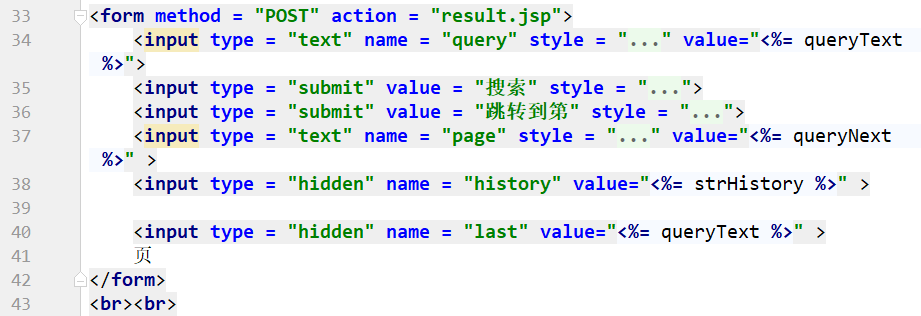
这部分还要考虑到搜索结果页面还可以继续搜索,

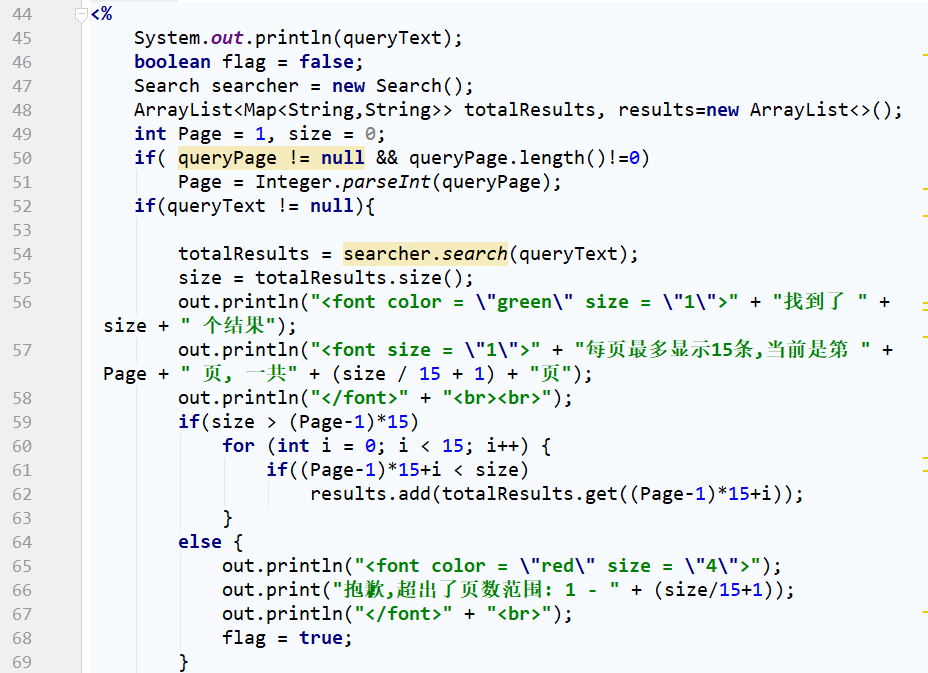
并且还要分页,跳转,储存搜索记录.

所以相比于初始界面要多一些东西.

这是开始部分的处理参数部分:

需要处理搜索串,页数,以及搜索记录.

这是本页面的post部分,包括查询字符串和跳转

这是后面输出的部分,包括调用Search类的查询函数,以及界面结果的格式化输出.

**【实验结果说明及演示】**

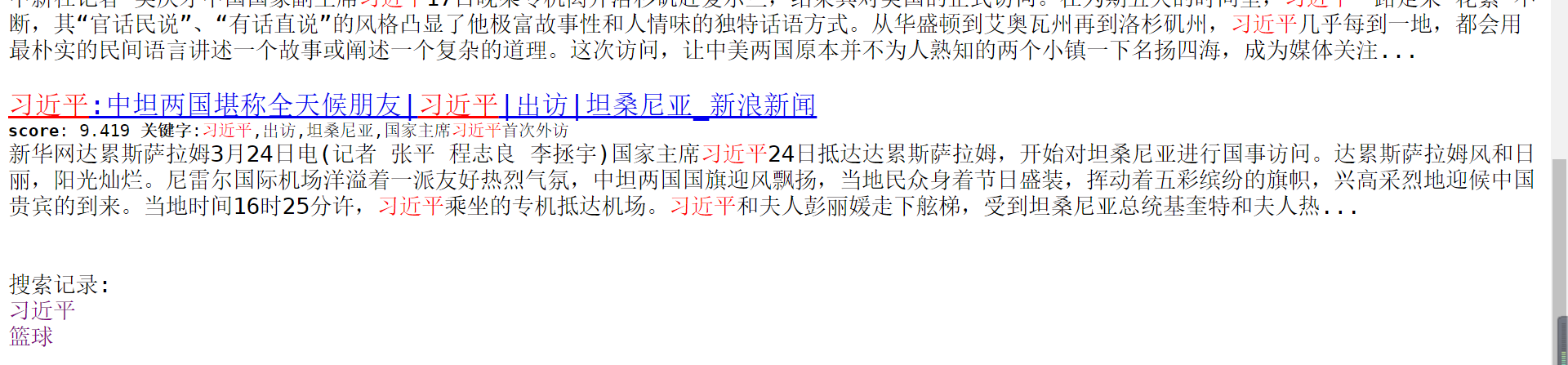
实现的功能: 高亮显示, 搜索记录, 结果分页.

初始页面:

搜索”篮球”的结果:

可以看出,有高亮,有分页,跳转.

 再搜索”习近平”:

 并且末尾有搜索记录:

 跳转到第20页:

 并且如果跳转页面非法,则会提示(容错性):

**【实验总结】**

**亮点:**

1. 使用的工具:

为了效果的极致,也为了让自己更贴近工业界,我使用的都是目前业界最广泛好评的工具,以及最新的版本.

比如IntelliJ IDEA这款比较新的IDE,个人认为大部分场合都完胜Eclipse.

比如Lucene 和 ICTCLAS,没有用FTP中的老版本,而是自己找的官方最新版.

1. 高性能:

创建索引的过程是在内存中全部完成之后再转移到磁盘中,这样大大加快了创建索引的过程.

1. 高亮:

高亮部分没有使用现成轮子,而是通过ICTCLAS的API,得到token流直接进行字符串匹配来替换.并且考虑到ICTCLAS分词工具偶尔出现的编码转换错误,进行了额外处理.

1. 分页:

没有用数据库进行分页的管理,而是通过简单的post传递,以及页数的计算进行分页显示.

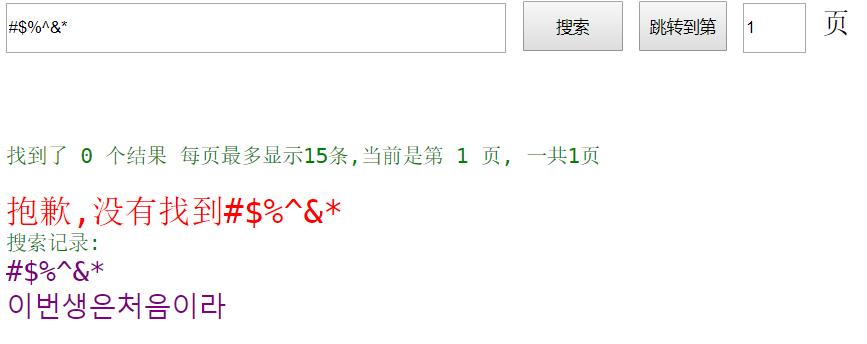
并且考虑到了超出页面限制的情况.

1. 搜索记录

同样没有使用数据库.而是使用一个数组来存储搜索记录,每次调用result时通过post传递,来进行显示.

1. 容错性

为了使最后的作品更加鲁棒,所以考虑了很多细节.

比如非法页数,非法字符等.

 **不足:**

1. 由于时间精力不足,没来得及实现推荐系统,高级搜索,模糊搜索等,这些如果加进来,那么会棒很多!
2. 性能还是有些不足,比如有个同学使用了多线程搜索,这一点可以改进.
3. 前端界面不是很美观,虽然也能看得过去.
4. 没有使用cookie,数据库,如果想要做大这个作品,就必须加这些东西.

以上就是不足和可以改进的地方!

总结:

这次实验的收获还是非常大的!借这个机会学习了java,

了解并学会掌握了lucene,ictclas,IntelliJ的使用,还学到了jsp.

自学能力再次提升!

虽然三个实验只占20分,并且没有多少区分度,但是本次试验的收获让我非常开心!

感谢!