

原

Lucene初试——关于大文本建立索引和中文乱码以及QueryParser检索的一些体会

2014年09月12日 10:35:49 sheen1991 阅读数：2564 标签： Analyzer Lucene全文检索 Lucene中文乱码 Lucene大文本 Lucene大文本索引

更多

版权声明：本文为博主原创文章，转载请注明出处。 [https://blog.csdn.net/shine\\_of\\_sun/article/details/39226113](https://blog.csdn.net/shine_of_sun/article/details/39226113)

这几天因为一个小项目用到Lucene，于是去学习了一下，现在还有很多地方没有了解，先就我遇到的问题做下总结。

一、大文本建索引问题

我这里说的大文本，实际上也就200M左右的txt，或许不应该成为大文本，但是我在建索引时遇到200M左右的的确导致了内存溢出，报错误 java.lang.OutOfMemoryError: Java heap space，到网上查了很久，试了一些方法，比如修改JVM的运行参数等，都不行。我测试的机器为i3740，4核可用内存1G多，按说对于200M的文本不应该可以接受吗？但是就是出现了内存溢出的情况。在对Lucene的机制还不了解的情况下，我想了一个方案，一个是切割文本，将大文本首先预处理以下，分成小的文本，二是在建立索引时，对于大文本分段建立，比如读到50M往磁盘写一次，一次读1w行。其实我觉得这些内在都是一个道理，就是一点点切分文本，只是实现方式稍有区别。

第一种情况我没有测试，觉得太麻烦，还要写个独立程序切割文本。第二种方式，我的代码逻辑是，读到一定大小的数据之后，就建立一个Document，设置setMaxBufferedDocs（n），我实测是10M的时候存一个Document，然后setMaxBufferedDocs（5），根据Lucene的官方文档，**当内存中的缓存到（我设置的200M）或者doc数目达到指定大小（也就是这里设置的5）时，就会触发一次往磁盘里写数据的操作**。我觉的这样的话，内存里顶多只会存不了几个IO太频繁降低速度而已，至少得能跑。但实际测试过程中，发现跑了很久很久都没有把一个文本（170M）索引建好，按照我的理解，170M也就作个几百万行了，但事实是很久没出现结果。于是我放弃了这种想法，没再去细究。

后来我用第三种方式测试了下，可行，而且效率还可以。思路是，每读1W行建立一个doc，我的这个文本比较特殊，2W行大概也就1m，然后设置setMaxBufferedDocs（100），这样反而可以。具体的数值设置可以根据自己环境来看，代码我也不贴了，就是循环按行读取文本，1w次之后建立一对象。

后来因为业务的需求，我又改成了1行建一个Document，然后setMaxBufferedDocs（2W），实测效率也可以。因为我检索时需要每一行的信息，所以取。比如我的一行数据为“1052307934---huajun7089059---73.63.134.205---安徽省滁州市电信ADSL---2011年7月10日---14:28:24”，我搜索“安徽省搜到了这条记录，那么我需要这一行的所有信息，如果不按每一行一个Document的话，比如我现在10行一个Document，那么噪音数据太多了。不知道有提供只提取我需要的信息的功能？要不然我就得自己写，从10行中找到我要的，那样也没有多大意义。

二、中文乱码问题

中文乱码真是无处不在。如果文件编码、系统编码、运行环境编码都一样应该不会出现乱码问题。如果知道文件的编码，一般

```
1 | reader = new BufferedReader(new InputStreamReader(  
2 |                                     new FileInputStream(file), "gbk"));
```

类似这样的设置就可以解决，但是我的比较麻烦，因为源文件有的是gbk有的是utf-8。所以还得动态的去识别当前文件的编码。我查了下，识别文件编码好像不是那么容易，例子都不是很精确或者稳定，还好我找了一个，对于我的txt还是有用的，代码如下，共享下（非原创）：

```
1 | /**  
2 |     * 查询字符编码  
3 |     * @param fileName  
4 |     * @return UTF-8/Unicode/UTF-16BE/GBK  
5 |     * @throws Exception  
6 |     */  
7 | public static String codeStringPlus(String fileName) throws Exception {  
8 |     BufferedInputStream bin = null;  
9 |     String code = null;  
10 |  
11 |     try {  
12 |         bin = new BufferedInputStream(new FileInputStream(fileName));  
13 |         int p = (bin.read() << 8) + bin.read();  
14 |         switch (p) {  
15 |             case 0xefbb:  
16 |                 code = "UTF-8";  
17 |                 break;  
18 |             case 0xfffe:  
19 |                 code = "Unicode";  
20 |                 break;  
21 |             case 0xffff:  
22 |                 code = "UTF-16BE";  
23 |                 break;  
24 |             default:  
25 |                 code = "GBK";  
26 |                 break;  
27 |         }  
28 |     } catch (Exception e) {  
29 |         e.printStackTrace();  
30 |     }  
31 | }
```

```
24 |         default: 25 |         code = "GBK"; 登录 注册 ×
26 |     }
27 | } catch (Exception e) {
28 |     e.printStackTrace();
29 | } finally {
30 |     bin.close();
31 | }
32 |
33 | return code;
34 | }
```

这样，我的编码问题就解决了。感谢这位贡献者，虽然可能还有缺陷，但是在我这够用了。

### 三、关于检索的一些认识

之前一直很困惑一个问题，场景如下，

比如文本“Hello , I am Chinese”，

建索引之后我查询“hello”、“Chinese”都是可以查到的，但我查询“Chine”为何就是查不到呢？难道是我写错了？

后来我想明白了，因为我查询的时候也用了分析器Analyzer。根据我的理解，我觉得Lucene是这样一个过程，在建立索引的时候首先分词，上述文本会分成“Hello”、“I”、“am”、“Chinese”，（当然可能I和am会被解析器去掉，这些价值不大，这里假设他们是有意义的）。

然后我查询的时候是用QueryParser加Analyzer查询的，也就是说，我输入Chinese的时候，首先也经历了分词的过程，会将我的关键字解析成“Chines”，是可以的，而当我输入“Chine”的时候，解析器只会解析成“Chine”，这个在索引里是没有的！当然查不到。如果想要输入Chine也能查到的话，可操作，比如换一种查询方式之类的，而我并没有去做，所以引起了这些困惑。

### 四、其他格式文本的检索

Lucene是不关心源文件的文件格式的，也就是说，得自己将不同格式的文档转换成纯文本，需要自己去写不同格式的解析器，而不是直接拿过来建索引。

以上是我目前对Lucene的理解，不知道是不是都对，仅供参考，希望读到此文的朋友能给出一点意见建议，共同学习。

### 五、下面附上代码：

环境是Windows下Lucene4.10.0+Myeclipse2013+JDK1.7，4GRam。

#### 1、建立索引（这里都是txt）

```
1  /**
2   * LuceneTest
3   * com.lucene.sheen.mine
4   */
5  package com.lucene.sheen.mine;
6
7  import java.io.BufferedInputStream;
8  import java.io.BufferedReader;
9  import java.io.File;
10 import java.io.FileInputStream;
11 import java.io.InputStreamReader;
12 import java.lang.management.ManagementFactory;
13 import java.lang.management.MemoryMXBean;
14 import java.lang.management.MemoryUsage;
15 import java.util.Date;
16
17 import org.apache.lucene.analysis.Analyzer;
18 import org.apache.lucene.document.Document;
19 import org.apache.lucene.document.Field;
20 import org.apache.lucene.document.Field.Store;
21 import org.apache.lucene.document.StringField;
22 import org.apache.lucene.document.TextField;
23 import org.apache.lucene.index.IndexWriter;
24 import org.apache.lucene.index.IndexWriterConfig;
25 import org.apache.lucene.index.IndexWriterConfig.OpenMode;
26 import org.apache.lucene.store.Directory;
27 import org.apache.lucene.store.FSDirectory;
```

```
30 import com.chenlb.mmseg4j.analysis.MMSegAnalyzer; 31 |
32 /**
33  * @author Sheen 2014-9-10
34  *
35  */
36 public class MyIndex {
37
38     /**
39      * @param args
40      * @throws Exception
41      */
42     public static void main(String[] args) throws Exception {
43         String docPath = "resource\\data";
44         String indexPath = "resource\\index";
45
46         File docFile = new File(docPath);
47         if (!docFile.exists() || !docFile.canRead()) {
48             System.out.println("您所选择的文件夹不存在或者没有访问权限! 文件路径: "
49                 + docFile.getAbsolutePath());
50             System.exit(1);
51         }
52         Date start = new Date();
53
54         Directory indexDir = FSDirectory.open(new File(indexPath));
55         Analyzer analyzer = new MMSegAnalyzer();
56         IndexWriterConfig iwc = new IndexWriterConfig(Version.LUCENE_4_10_0,
57             analyzer);
58         iwc.setRAMBufferSizeMB(200).setMaxBufferedDocs(20000);
59         iwc.setOpenMode(OpenMode.CREATE);
60         IndexWriter writer = new IndexWriter(indexDir, iwc);
61
62         MemoryMXBean memorymbean = ManagementFactory.getMemoryMXBean();
63         MemoryUsage usage = memorymbean.getHeapMemoryUsage();
64         System.out.println("INIT HEAP: " + usage.getInit());
65         System.out.println("MAX HEAP: " + usage.getMax());
66         System.out.println("USE HEAP: " + usage.getUsed());
67
68         indexDoc(writer, docFile);
69         writer.close();
70         Date end = new Date();
71         seeVMStatus();
72         System.out.println("所有文件建立索引完毕, 耗时: "
73             + (double) (end.getTime() - start.getTime()) / (1000 * 60)
74             + "min");
75     }
76
77     static void indexDoc(IndexWriter writer, File file) throws Exception {
78         if (file.canRead()) {
79             if (file.isDirectory()) {
80                 File[] files = file.listFiles();
81                 for (File thisFile : files) {
82                     indexDoc(writer, thisFile);
83                 }
84             } else {
85                 String code = codeString(file.getAbsolutePath());
86                 System.out.println("*****文件: " + file.getAbsolutePath()
87                     + "正在建立索引*****");
88                 System.out.println("字符编码: " + code);
89                 seeVMStatus();
90                 BufferedReader reader = null;
91                 try {
92                     Field pathField = new StringField("path", file.getPath(),
93                         Field.Store.YES);
94                     reader = new BufferedReader(new InputStreamReader(
95                         new FileInputStream(file), code));
96                     String line = null;
97                     long fileSize = 0;
98                     while ((line = reader.readLine()) != null) {
99                         fileSize += line.getBytes().length;
100                         Document doc = new Document();
101                         doc.add(pathField);
```

```

102 |                                     Field textField = new TextField("contents", line,
|                                     Store.YES);104 |
|                                     doc.add(textField);105 |
|                                     writer.addDocument(doc);106 |
107 |                                     }
108 |                                     System.out.println("TotalSize:" + fileSize / (1024 * 1024)
109 |                                     + "M");
110 |                                     System.out.println("建立索引完毕\n");
111 |
112 |                                     } catch (Exception e) {
113 |                                         e.printStackTrace();
114 |                                     } finally {
115 |                                         reader.close();
116 |                                     }
117 |                                 }
118 |
119 |                             }
120 |                         }
121 |
122 | /**
123 |  * 查看虚拟机内存信息
124 |  */
125 | public static void seeVMStatus() {
126 |     MemoryMXBean memorymbean = ManagementFactory.getMemoryMXBean();
127 |     System.out.println("JVM Full Information:");
128 |     System.out.println("Heap Memory Usage: "
129 |         + memorymbean.getHeapMemoryUsage());
130 |     System.out.println("Non-Heap Memory Usage: "
131 |         + memorymbean.getNonHeapMemoryUsage());
132 | }
133 |
134 |
135 | /**
136 |  * 查询字符编码
137 |  * @param fileName
138 |  * @return UTF-8/Unicode/UTF-16BE/GBK
139 |  * @throws Exception
140 |  */
141 | public static String codeStringPlus(String fileName) throws Exception {
142 |     BufferedInputStream bin = null;
143 |     String code = null;
144 |
145 |     try {
146 |         bin = new BufferedInputStream(new FileInputStream(fileName));
147 |         int p = (bin.read() << 8) + bin.read();
148 |         switch (p) {
149 |             case 0xefbb:
150 |                 code = "UTF-8";
151 |                 break;
152 |             case 0xfffe:
153 |                 code = "Unicode";
154 |                 break;
155 |             case 0xfeff:
156 |                 code = "UTF-16BE";
157 |                 break;
158 |             default:
159 |                 code = "GBK";
160 |         }
161 |     } catch (Exception e) {
162 |         e.printStackTrace();
163 |     } finally {
164 |         bin.close();
165 |     }
166 |
167 |     return code;
168 | }
169 | /**
170 |  * 查询字符编码是UTF-8还是GBK
171 |  * @param fileName
172 |  * @return UTF-8/GBK

```

```

173 |         * @throws Exception 174 |         */
175 |         public static String codeString(String fileName) throws Exception {
176 |             BufferedInputStream bin = null;
177 |             String code = null;
178 |             try {
179 |                 bin = new BufferedInputStream(new FileInputStream(fileName));
180 |                 int p = (bin.read() << 8) + bin.read();
181 |                 switch (p) {
182 |                     case 0xefbb:
183 |                         code = "UTF-8";
184 |                         break;
185 |                     default:
186 |                         code = "GBK";
187 |                 }
188 |             } catch (Exception e) {
189 |                 e.printStackTrace();
190 |             } finally {
191 |                 bin.close();
192 |             }
193 |             return code;
194 |         }
195 |
196 |
197 | }

```

## 2、查询

```

1 | /**
2 |  * LuceneTest
3 |  * com.lucene.sheen.mine
4 |  */
5 | package com.lucene.sheen.mine;
6 |
7 | import java.io.File;
8 | import java.io.IOException;
9 | import java.util.Date;
10 |
11 | import org.apache.lucene.analysis.Analyzer;
12 | import org.apache.lucene.document.Document;
13 | import org.apache.lucene.index.DirectoryReader;
14 | import org.apache.lucene.index.IndexReader;
15 | import org.apache.lucene.queryparser.classic.ParseException;
16 | import org.apache.lucene.queryparser.classic.QueryParser;
17 | import org.apache.lucene.search.IndexSearcher;
18 | import org.apache.lucene.search.Query;
19 | import org.apache.lucene.search.ScoreDoc;
20 | import org.apache.lucene.search.TopDocs;
21 | import org.apache.lucene.store.FSDirectory;
22 |
23 | import com.chenlb.mmseg4j.analysis.MMSegAnalyzer;
24 |
25 | /**
26 |  * @author Sheen 2014-9-10
27 |  *
28 |  */
29 | public class MySearcher {
30 |
31 |     /**
32 |     * @param args
33 |     * @throws IOException
34 |     * @throws ParseException
35 |     */
36 |     public static void main(String[] args) throws IOException, ParseException {
37 |         String index = "resource\\index";
38 |         String field = "contents";
39 |         String queryString = "870270291";
40 |
41 |         IndexReader reader = DirectoryReader.open(FSDirectory.open(new File(index)));
42 |         IndexSearcher searcher = new IndexSearcher(reader);

```


```
43 |
44 |             Analyzer analyzer = new MMSegAnalyzer();
45 |
46 |             QueryParser parser = new QueryParser(field, analyzer);
47 |             Query query = parser.parse(queryString);
48 |             System.out.println("查询关键字: "+query.toString());
49 |             Date start = new Date();
50 |             TopDocs results = searcher.search(query, 20);
51 |             ScoreDoc[] hits = results.scoreDocs;
52 |             for(ScoreDoc sdoc : hits){
53 |                 Document doc = searcher.doc(sdoc.doc);
54 |                 System.out.println("查询结果: ");
55 |                 System.out.println(sdoc.score);
56 |                 System.out.println(doc.get("path"));
57 |                 System.out.println(new String(doc.get("contents").getBytes(),"UTF-8"));
58 |             }
59 |             Date end = new Date();
60 |             System.out.println("耗时: "+(end.getTime()-start.getTime()));
61 |         }
62 |
63 |     }
```

Python爬虫全栈教学，零基础教你成编程大神

零基础学爬虫，你要掌握学习那些技能？

想对作者说点什么？

我来说两句

 Dolphin-J：拜读！（2年前 #1楼）

- lucene-检索文本内容失败问题

问题描述：①通过创建索引库，将源文件夹中的文件，通过document文档对象，创建索引存储在索引... 来自： gsy\_yang的博客

755
- Lucene学习总结之三：Lucene的索引文件格式(1)

Lucene的索引里面存了些什么，如何存放的，也即Lucene的索引文件格式，是读懂Lucene源代码的... 来自： jediael\_lu的专栏

3473
- lucene的建立索引，搜索，中文分词

Lucene是apache软件基金会4 jakarta项目组的一个子项目，是一个开放源代码的全文检索引擎工具包... 来自： young\_so\_nice的博客

3150
- 30W年薪的人工智能工程师只是“白菜价”？

机器学习|深度学习|图像处理|自然语言处理|无人驾驶，这些技术都会吗？看看真正的人工智能师都会那些关键技术？年薪比你高多少！
- 不同规则的中文分词对Lucene索引的影响

在中文全文索引中为了建立反向索引需要对文档中的句子进行切分，相关理论请参见车东的介绍。在l... 来自： Jason的专栏

158
- Lucene 需要索引的文本文件太大,怎么解决？

2010-04-01 16:34有问题的过路人 | 分类： JAVA相关 | 浏览1599次 我用的lucene版本是3.0.1,当我索... 来自： Java/Android/iOS/前...

806
- 对中文文本生成的索引文件中有乱码

原因可能是中文的文本文件的编码格式与项目的编码格式不一致造成的。比如我的项目中Java文件的... 来自： wangchenggong198...

818
- 写入文件以及解决文件名中文乱码问题

public ModelAndView expSequences(String id, HttpServletRequest request, HttpServletResponse re... 来自： liuzhen917的专栏

616
- 利用lucene对文件内容进行关键字检索

一、概述 关于lucene的具体介绍，请自行百度。 二、实例讲解 在具体实现之前，请根据自... 来自： zh\_winer的博客

1733

## Lucene学习笔记之（六）lucene的中文查询

步骤一：创建maven项目 步骤二：配置pom.xml文件 junit junit ...

4719

来自：孤师的博客

## 全文检索技术学习(三)——Lucene支持中文分词

分析器（Analyzer）的执行过程如下图是语汇单元的生成过程：从一个Reader字符流开始，创建一个...

7377

来自：李阿昀的博客



xiaojimanman  
关注 98篇文章



刘迎光-萤火虫工作室  
关注 256篇文章



jerome\_s  
关注 295篇文章

## Lucene学习笔记之（一）简介和向文档写索引并读取文档

什么是lucene? lucene的官方文档http://lucene.apache.org/core/ lucene就是一个全文检索的工具包。 ...

5754

来自：孤师的博客

## Lucene建立索引及查询包含“java”关键字 示例代码

根据博客调试的程序，比较简单，英文分词和检索，希望对大家学习有帮助。

## lucene索引

Lucene介绍：注意：在建立索引搜索时如果中间有乱码的话，是没有效果的 后续会增加遍历源文件...

122

来自：y666666y的博客

## 陈小春坦言：这游戏不充钱都能当全服大哥，找到充值入口算我输！

贪玩游戏 · 顶新

## Lucene 实现txt文件的构建索引与查询

```
package net.jqsoft.hecv.util; import net.sf.json.JSONArray; import org.apache.lucene.analysis.Analy...
```

791

来自：@/"蜗牛

## java之全文索引搜索lucene之增删改查文档与中文分词搜索

介绍一下lucene的各种搜索，即query以及中文分析索引搜索与对索引结果高亮显示...

3213

来自：steadyjack博客

## Lucene之超链接传中文乱码

最近学习ssm使用到Lucene来进行全文检索，取代了以前的模糊查询，与使用like来模糊查询Lucene... 来自：weixin\_37762923的...

240

## JAWAWEB开发之Lucene详解——Lucene入门及使用场景、全文检索、索引CRUD、优化索...

Lucene入门 应用场景 windows系统中的有搜索功能：打开“我的电脑”，按“F3”就可以使用查找的功能...

1883

## 全文检索技术学习(一)——Lucene的介绍

本文我将为大家讲解全文检索技术——Lucene，现在这个技术用到的比较多，我觉得大家还是应该掌...

1994

来自：李阿昀的博客



全球LED显示屏生产厂家排名

百度广告

## solr全文检索实现原理

solr那是我1年前使用到的一个搜索引擎，由于当初对于配置了相应了，但是今天突然面试问到了，哎... 来自：一个菜鸟的学习之路

2.7万

## html抽取文本信息-java版（适合lucene建立索引）

```
import org.htmlparser.NodeFilter; import org.htmlparser.Parser; import org.htmlparser.beans.StringBe... 来自：落叶翩翩的CSDN博客
```

1406

## lucene建立pdf文件内容全文索引

第一次写博客，有很多不足，也许词不达意，也许有其他问题，作为一个新手，第一次使用lucene， ...

305

## 基于lucene的案例开发：Query查询

在Lucene索引的搜索过程中，构建Query对象是一个十分重要的过程，在这篇博客中，我们将重点介... 来自：小鸡慢慢的专栏

4073

## lucene查询之 MultFieldQueryParser 可以指定多个默认搜索域；及代码示例；代码实现

1.1.1. MultFieldQueryParser 可以指定多个默认搜索域 @Test public void testMultiFiledQueryParser()... 来自：boss\_way的博客

498

## 你卸载算我输系列！爆率9.8，要充值要VIP算我输！

贪玩游戏 · 顶新



- 【Lucene4.8教程之六】QueryParser与Query子类：如何生成Query对象

对于一个搜索而言，其核心语句为： searcher.search(query, 10);此时，其最重要的参数为一个Query...

来自： jediael\_lu的专栏

9388
- Lucene中的highlighter

Lucene中的highlighter除了高亮关键词之外还有动态摘要的功能.动态摘要就是将文本分成若干片段(fr...

来自： jollyjumper的专栏

1846
- Lucene(Lucene)建立索引(字段)

Lucene,这是官方称谓,也有许多人叫它Lucence,做搜索和分词用的工具包.也有人说是Java下的搜索引...

来自： LightSun----让成功成...

1626
- Clucene 文件数据读取

在这里给大家分享一下基于Clucene 索引数据读取程序。该程序是基于VC++，使用Clucene 0.92 版...

来自： dongjian764的专栏

511
- lucene3.1.0+luke3.5.0开发中遇到的问题解决办法

在虚拟机上解压完lucene3.1.0后，运行luke(基于lucene的查看索引的第三方工具)，运行命令 #java ...

来自： lvyuan1234的博客

675
-  开发一个app大概需要多少钱呢

百度广告
- Lucene建立索引库

问题？Lucene如何建立索引库， lucene所需要的jar包是那些， lucene如何使用索引库， lucene的核...

来自： Mr\_li13的博客

5059
- Lucene索引并检索数据库

比较深入的讲解了Lucene索引和检索原理，以及如何对数据库建立索引和检索数据...

来自： 言宜慢，心宜善

4034
- 关于Lucene以及索引和搜索的流程

Lucene的普及和成功的背后是因为它的简单。 因此，你不需要深入理解Lucene的信息索引和检索...

来自： 算法魔功

1896
- 10、索引库的查询四之：Lucene的高级搜索技术

Lucene的高级搜索技术首先要说的就是 SpanTermQuery，他和TermQuery用法很相似，唯一区别就...

来自： u013115157的博客

1549
- Lucene系列 - 索引(七) - 对数据库记录建立索引

数据库本就具有检索功能，使用标准的Sql语句很容易从一个表中获取指定的内容。 另一方面，越来...

来自： 大泡泡的专栏

2125
- 我们对天发誓：这正版传奇爆率9.8，送VIP，卸载算我输！

贪玩游戏 · 顶新
- Lucene创建索引和索引的基本检索

Lucene创建索引和索引的基本检索 Author： 百知教育gaozhy 注：演示代码所使用jar包版本为 lucene...

来自： 高興

312
- Lucene查询索引

Lucene简介和怎样创建索引可以参考我前面的两篇博客Lucene全文检索基础和Lucene创建索引，索...

来自： 1.01^365=37.78 (Luc...

3801
- Lucene索引创建方法和步骤

在全文索引工具中，都是由这样的三部分组成 1.索引部分 2.分词部分 3.搜索部分 -----...

来自： Jonney's house

1352
- lucene-亚洲语种和unicode字符分析

1、unicode字符在lucene内部，所有的字符都是以标准UTF-8编码的，JAVA会在字符串对象内对Unic...

来自： 深未来技术

512
- Lucene整理--索引的建立

创建索引的过程如下：（1）、建立索引器IndexWriter，这相当于一本书的框架（2）、建立文档对...

来自： 曹海成的专栏

3471
- 你卸载算我输系列！爆率9.8，要充值要VIP算我输！

贪玩游戏 · 顶新
- lucene3.0的查询语法和QueryParser的语法

Query有很多子类，完成不同类型的查询任务： Instantiable subclasses are: TermQueryMultiTermQu...

来自： 云守护的专栏

4420



- Lucene 学习（三）：在一个(或者多个)字段中查找多个关键字

上一篇中，实现了中文分词的操作，在实际试用中，发现众多搜索引擎的搜索框中，我们在不同的key... 来自：The magic of fingertips
- Lucene简单实现创建索引以及查询

package com.rdz.test; import java.io.File; import java.io.FileReader; import java.io.IOException; ... 来自：甲如没有乙后的博客
- lucene全文搜索之三：生成索引字段，创建索引文档（给索引字段加权）基于lucene5.5.3

前言：上一章中我们已经实现了索引器的创建，但是我们没有索引文档，本章将会讲解如何生成字段... 来自：做好自己！- eguid
- lucene 高级搜索

出处：http://www.cnblogs.com/kissdodog/p/3640349.html 感谢您的文章 一、解析搜索请求 搜索... 来自：青衫-CSDN博客



项目管理 软件

百度广告

- lucene学习 内存索引库和文件索引库结合

在lucene索引库的创建的时候，我们有两种不同的索引库创建方式 1.文件索引库final Path docDir = P... 来自：学习小窝
- 加了中文分词后，就查询不到任何东东了

加了中文分词后，就查询不到任何东东了，我用Luke工具查询了索引文件，发现索引完全正常，而且... 来自：moonlight的专栏
- Flask用reqparse实现post方法时，parser.parse\_args()的值为None的解决办法

Flask—restful 的restful API实现from flask import Flask from flask\_restful import Api, Resource, reqpa... 来自：老鼠程序员的专栏
- 用于Lucene的各中文分词比较

对几种中文分析器，从分词准确性和效率两方面进行比较。分析器依次为：StandardAnalyzer、Chine... 来自：程序猿开发日志【学...
- lucene+compass中文乱码问题

大家谁碰到过，我感觉是个常见问题。有那位解决的！ 来自：GODOMONEY的专栏
- BIM施工技术简介

百度广告
- lucene 的使用详情

这是一篇关于lucene的讲解非常细的一篇文章，有助于大家对lucene更快的了解，... 来自：u011913691的博客



sheen1991

关注

原创38

粉丝20

喜欢3

评论17

等级：博客4

访问：9万+

积分：1339

排名：4万+



数据可视化工具

最新文章

Strom 消息可靠性保障机制和Ack原理

Storm学习小结(二)——集成JDBC和Redis

C3P0连接池小记

Maven入门笔记

storm学习笔记

个人分类

Storm

3篇

Java基础知识

5篇

SSH/SSI

11篇

数据库相关

5篇

Web开发小结

7篇

展开

归档

2016年10月

1篇

2016年8月

1篇

2016年7月

2篇

2016年6月

1篇

2015年8月

2篇

展开

热门文章

Windows下使用VNC连接CentOS7远程桌面

阅读量：12975

CentOS7\_x86\_64安装Oracle10g R2血泪史

阅读量：8094

CentOS7\_x86\_64部署WebLogic11g并设置开机自启

阅读量：6644

CentOS7重置root密码修改主机名

阅读量：6245

storm学习笔记

阅读量：5269

最新评论

CentOS7\_x86\_64安装O...

qq\_43194535：感谢！过程非常详细！很受用

Java Web文件下载1——文件...

lc\_stone：String.format("attachment; filename=\"%...

CentOS7重置root密码修改...

e8love：ko

Strom 消息可靠性保障机制和A...

jiabinzhang4310：楼主分析的太透彻了，膜拜。。。

CentOS7重置root密码修改...

Blue\_millet：passwd



### 人脸识别算法



#### 联系我们



扫码联系客服



下载CSDN APP

 QQ客服

 kefu@csdn.net

 客服论坛

 400-660-0108

工作时间 8:00-22:00

关于我们 招聘 广告服务 网站地图

 百度提供站内搜索 京ICP证09002463号

©2018 CSDN版权所有

网络110报警服务 经营性网站备案信息

北京互联网违法和不良信息举报中心

中国互联网举报中心