全文检索技术学习(三)——Lucene支持中文分词

05月21日 00:23:04 李阿昀 阅读数：9984

析器（Analyzer）的执行过程

是语汇单元的生成过程：

Reader字符流开始，创建一个基于Reader的Tokenizer分词器，经过三个TokenFilter生成语汇单元Token。
析器的分析效果，只需要看TokenStream中的内容就可以了。每个分析器都有一个方法tokenStream，返回的是一个TokenStream对象。

准分析器的分词效果

们创建索引库的时候，就用到了官方推荐的标准分析器——org.apache.lucene.analysis.standard.StandardAnalyzer。现在我们就来看看其分词效果，
eFirst单元测试类中编写如下方法：

```
public class LuenceFirst {

    // 查看分析器的分词效果
    @Test
    public void testAnanlyzer() throws IOException {
        // 1、创建一个分析器对象
        Analyzer analyzer = new StandardAnalyzer(); // 官方推荐的标准分析器
        // 2、从分析器对象中获得tokenStream对象
        // 参数1：域的名称，可以为null，或者是""
        // 参数2：要分析的文本
        TokenStream tokenStream = analyzer.tokenStream("", "The Spring Framework provides a comprehensive programming and configuration model.");

        // 3、设置一个引用(相当于指针)，这个引用可以是多种类型，可以是关键词的引用，偏移量的引用等等
        CharTermAttribute charTermAttribute = tokenStream.addAttribute(CharTermAttribute.class); // charTermAttribute对象代表当前的关键
        // 偏移量(其实就是关键词在文档中出现的位置，拿到这个位置有什么用呢？因为我们将来可能要对该关键词进行高亮显示，进行高亮显示要知道这个关键
        OffsetAttribute offsetAttribute = tokenStream.addAttribute(OffsetAttribute.class);
        // 4、调用tokenStream的reset方法，不调用该方法，会抛出一个异常
        tokenStream.reset();
        // 5、使用while循环来遍历单词列表
        while (tokenStream.incrementToken()) {
            System.out.println("start→" + offsetAttribute.startOffset()); // 关键词起始位置
            // 6、打印单词
            System.out.println(charTermAttribute);
            System.out.println("end→" + offsetAttribute.endOffset()); // 关键词结束位置
        }
        // 7、关闭tokenStream对象
        tokenStream.close();
    }

}
```

上方法，Eclipse控制台打印：


```
rt→4
ing
→10
rt→11
mework
→20
rt→21
vides
→29
rt→32
prehensive
→45
rt→46
gramming
→57
rt→62
figuration
→75
rt→76
el
→81
```

中我们可以清楚地看到当前的关键词，以及该关键词的起始位置和结束位置。

分析器分析

ene自带中文分词器

e自带的中文分词器有：

andardAnalyzer
字分词，就是按照中文一个字一个字地进行分词。如：“我爱中国”，效果：“我”、“爱”、“中”、“国”。

JKAnalyzer
分法分词，按两个字进行切分。如：“我是中国人”，效果：“我是”、“是中”、“中国”、“国人”。
边这两个分词器一看就无法满足需求。

martChineseAnalyzer
中文支持较好，但扩展性差，扩展词库，禁用词库和同义词库等不好处理。

们来看看第三个中文分析器的分析效果，相比前两个中文分析器，SmartChineseAnalyzer绝对要胜出一筹。为了观看其分析效果，我们可将Luen st Ananlyzer方法改造为：

```
public class LuenceFirst {

    // 查看分析器的分词效果
    @Test
    public void testAnanlyzer() throws IOException {
        // 1、创建一个分析器对象
        Analyzer analyzer = new SmartChineseAnalyzer(); // 智能中文分析器
        // 2、从分析器对象中获得tokenStream对象
        // 参数1：域的名称，可以为null，或者是""
        // 参数2：要分析的文本
        TokenStream tokenStream = analyzer.tokenStream("", "数据库中存储的数据是结构化数据，即行数据java，可以用二维表结构来逻辑表达实现的数据。");
```

```
        CharTermAttribute charTermAttribute = tokenStream.addAttribute(CharTermAttribute.class); // cha                    uti                    前的关键词
        // 偏移量(其实就是关键词在文档中出现的位置，拿到这个位置有什么用呢？因为我们将来可能要对该关键词进行高亮                                                    个关键词在哪？,
        OffsetAttribute offsetAttribute = tokenStream.addAttribute(OffsetAttribute.class);
        // 4、调用tokenStream的reset方法，不调用该方法，会抛出一个异常
        tokenStream.reset();
        // 5、使用while循环来遍历单词列表
        while (tokenStream.incrementToken()) {
            System.out.println("start→" + offsetAttribute.startOffset()); // 关键词起始位置
            // 6、打印单词
            System.out.println(charTermAttribute);
            System.out.println("end→" + offsetAttribute.endOffset()); // 关键词结束位置
        }
        // 7、关闭tokenStream对象
        tokenStream.close();
    }

}
```

上方法，Eclipse控制台打印：

```
rt→0
居库
→3
rt→3

→4
rt→4
者
→6
rt→6

→7
rt→7
居
→9
rt→9

→10
rt→10
句
→12
rt→12

→13
rt→13
居
→15
rt→16

→17
rt→17
```

```
rt→18
居
→20
rt→20
a
→24
rt→25
人
→27
rt→27

→28
rt→28

→29
rt→29

→30
rt→30

→31
rt→31
匀
→33
rt→33

→34
rt→34
事
→36
rt→36
左
→38
rt→38
见
→40
rt→40

→41
rt→41
居
→43
```

martChineseAnalyzer分析器对中文支持较好，但扩展性差，扩展词库，禁用词库和同义词库等不好处理。故实际开发中我们也是弃用的，取而代之的是

方由文分析器

aoding：庖丁解牛最新版在https://code.google.com/p/paoding/ ，其最多只支持Lucene3.0，且最新提交的代 登录 06 注册 中最新也是20
经过时，不予考虑。

mseg4j：最新版已从https://code.google.com/p/mmseg4j/移至https://github.com/chenlb/mmseg4j-solr，支持Lucene4.10，且在github中最新
014年6月，从09年～14年一共有18个版本，也就是一年几乎有3个大小版本，有较大的活跃度，用了mmseg算法。

-analyzer：最新版在https://code.google.com/p/ik-analyzer/上，支持Lucene4.10，从2006年12月推出1.0版开始，IKAnalyzer已经推出了4个大版本
以开源项目Luence为应用主体的，结合词典分词和文法分析算法的中文分词组件。从3.0版本开始，IK发展为面向Java的公用分词组件，独立于Lucene项
Lucene的默认优化实现。在2012版本中，IK实现了简单的分词歧义排除算法，标志着IK分词器从单纯的词典分词向模拟语义分词衍化。但是也就在2012
更新了。

nsj_seg：最新版本在https://github.com/NLPchina/ansj_seg tags，仅有1.1版本，从2012年到2014
次，但是作者本人在2014年10月10日说明：“可能我以后没有精力来维护ansj_seg了”，现在由”nlp_china”管理。2014年11月有更新。并未说明是否
ucene，是一个由CRF（条件随机场）算法所做的分词算法。

ndict-chinese-analyzer：最新版在https://code.google.com/p/imdict-chinese-analyzer/，最新更新也在2009年5月，可下载源码，不支持Lucene4.1
MM（隐马尔科夫链）算法。

seg：最新版本在git.oschina.net/lionsoul/jcseg，支持Lucene 4.10，作者有较高的活跃度。其利用的是mmseg算法。

里，我使用的是IK-analyzer，所以下面的讲解也是围绕着该中文分析器来进行的。下面是我下载的IK-analyzer：

Analyzer 2012FF_hf1.zip

之后，其目录结构是：

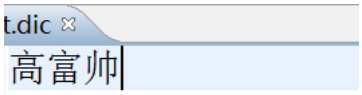
nalyzer中文分析器的使用

alyzer中文分析器的使用步骤：

把IKAnalyzer2012FF_u1.jar包添加到工程中。

把配置文件和扩展词典和停用词词典添加到classpath下。

扩展词典和停用词词典这两个文件的字符集一定要保证是UTF-8字符集，注意是无BOM的UTF-8编码，严禁使用Windows的记事本编辑。
们来看看IK-analyzer这个第三方中文分析器的分析效果。现在随着互联网的日趋发展，网络用语层出不穷，例如“高富帅”，“白富美”等等，
行分词的，而是当作一个整体的关键词，这样像这种不用分词的网络用语就应该存储在扩展词典中。为了清楚地观看IK-analyzer这个第三方中文
词典添加“高富帅”。如下：



LuenceFirst单元测试类中的testAnanlyzer方法改造为：

```
public class LuenceFirst {
```

```
    public void testAnanlyzer() throws IOException {
        //1、创建一个分析器对象
        Analyzer analyzer = new IKAnalyzer(); //智能中文分析器
        //2、从分析器对象中获得tokenStream对象
        //参数1：域的名称，可以为null，或者是""
        //参数2：要分析的文本
        TokenStream tokenStream = analyzer.tokenStream("", "数据库中存储的数据是结构化数据高富帅，即行数据java，可以用二维表结构来逻辑表达实现的数据

        //3、设置一个引用(相当于指针)，这个引用可以是多种类型，可以是关键词的引用，偏移量的引用等等
        CharTermAttribute charTermAttribute = tokenStream.addAttribute(CharTermAttribute.class); //charTermAttribute对象代表当前的关键词
        //偏移量(其实就是关键词在文档中出现的位置，拿到这个位置有什么用呢？因为我们将来可能要对该关键词进行高亮显示，进行高亮显示要知道这个关键词在哪？,
        OffsetAttribute offsetAttribute = tokenStream.addAttribute(OffsetAttribute.class);
        //4、调用tokenStream的reset方法，不调用该方法，会抛出一个异常
        tokenStream.reset();
        //5、使用while循环来遍历单词列表
        while (tokenStream.incrementToken()) {
            System.out.println("start→" + offsetAttribute.startOffset()); //关键词起始位置
            //6、打印单词
            System.out.println(charTermAttribute);
            System.out.println("end→" + offsetAttribute.endOffset()); //关键词结束位置
        }
        //7、关闭tokenStream对象
        tokenStream.close();
    }

}
```

上方法，Eclipse控制台打印：

```
戈扩展词典：ext.dic
戈扩展停止词典：stopword.dic
rt→0
据库
→3
rt→0
据
→2
rt→2

→3
rt→3

→4
rt→4
者
→6
rt→7
据
→9
rt→9

→10
rt→10
构化
```

习
→12
rt→12

→13
rt→13
居
→15
rt→15
富帅
→18
rt→19
于
→21
rt→20
女
→22
rt→21
居
→23
rt→23
a
→27
rt→28
人用
→31
rt→28
人
→30
rt→30

→31
rt→31
生
→33
rt→31

→32
rt→32

→33
rt→33

→34
rt→34
勾

```
→37
rt→37
→39
rt→39
→41
rt→41
→43
rt→44
→46
```


可清楚地看出“高富帅”并没有分词，这正是我们所期望的结果。
外，对于一些敏感的词，如“习近平”，像这样的敏感词汇就不应该出现在单词列表中，所以可将这种敏感词汇存储在停用词词典中，如下：

stopword.dic

```
such
that
the
their
then
there
these
they
this
to
was
will
with
的
了
习近平
```

LuenceFirst单元测试类中的testAnanlyzer方法改造为：

```
public class LuenceFirst {

    //查看分析器的分词效果
    @Test
    public void testAnanlyzer() throws IOException {
        //1、创建一个分析器对象
        Analyzer analyzer = new IKAnalyzer(); //智能中文分析器
        //2、从分析器对象中获得tokenStream对象
        //参数1：域的名称，可以为null，或者是""
        //参数2：要分析的文本
        TokenStream tokenStream = analyzer.tokenStream("", "数据库习近平中存储的数据是结构化数据高富帅，即行数据java，可以用二维表结构来逻辑表达实现

        //3、设置一个引用(相当于指针)，这个引用可以是多种类型，可以是关键词的引用，偏移量的引用等等
```

```
        //4、调用tokenStream的reset方法，不调用该方法，会抛出一个异常
        tokenStream.reset();
        //5、使用while循环来遍历单词列表
        while (tokenStream.incrementToken()) {
            System.out.println("start→" + offsetAttribute.startOffset()); //关键词起始位置
            //6、打印单词
            System.out.println(charTermAttribute);
            System.out.println("end→" + offsetAttribute.endOffset()); //关键词结束位置
        }
        //7、关闭tokenStream对象
        tokenStream.close();
    }

}
```

上方法，Eclipse控制台打印：

```
我扩展词典：ext.dic
我扩展停止词典：stopword.dic
art→0
居库
l→3
art→0
居
l→2
art→2

l→3
art→6
```


可知，像“习近平”这样的敏感词汇并没有出现在单词列表中。

器的应用场景

时使用Analyzer

键字进行搜索，当需要让该关键字与文档域内容所包含的词进行匹配时需要对文档域内容进行分析，需要经过Analyzer分析器处理生成语汇单元（n
象是文档中的Field域。当Field的属性tokenized（是否分词）为true时会对Field值进行分析，如下图：

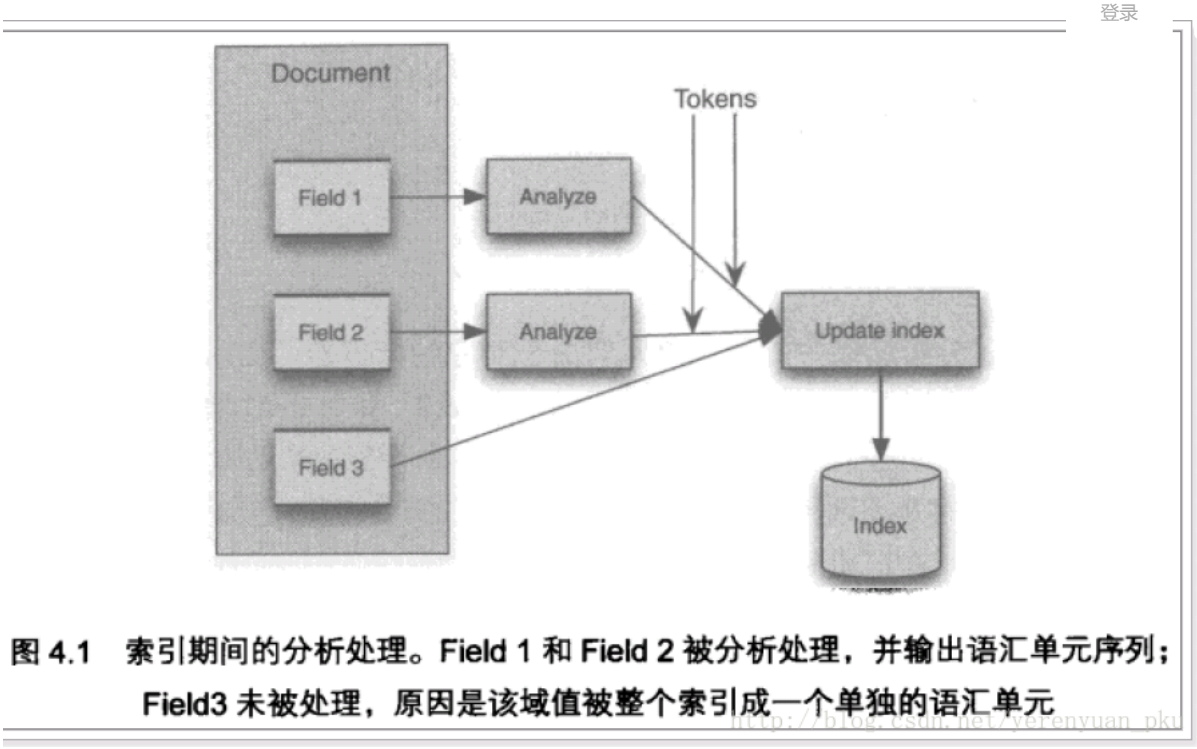


图 4.1 索引期间的分析处理。Field 1 和 Field 2 被分析处理，并输出语汇单元序列；Field3 未被处理，原因是该域值被整个索引成一个单独的语汇单元

些Field可以不用分析：

不作为查询条件的内容，比如文件路径

不是匹配内容中的词而匹配Field的整体内容，比如订单号、身份证号等

时使用Analyzer

关键字进行分析和索引分析一样，使用Analyzer对搜索关键字进行分析、分词处理，使用分析后的每个词语进行搜索。比如：搜索关键字：spring web，词，得出：spring web，拿词去索引词典表查找 ，找到索引链接到Document，解析Document内容。

配整体Field域的查询可以在搜索时不分析，比如根据订单号、身份证号查询等。

搜索使用的分析器要和索引使用的分析器最好保持一致。

ne实现中文分词 阅读数 3C
的文章中已经介绍过Lucene了，这里就不多做介绍。一、中文分词的原理中文分词是将一个汉字... 博文 来自： jhonve的博客

ne几种中文分词的总结 阅读数 30
rl]http://blog.sina.com.cn/s/print_4ff5925f01000d32.html[/url]内容:目前最新版本的lucene... 博文 来自： neptune

ne之中文分词器 阅读数 141
在lucene4.6.0以上版本使用IKAnalyzer时可能会出现以下异常：java.lang.illegalstateexceptio... 博文 来自： 张育嘉的博客

下的中文分词IKAnalyzerNet(基于Lucene.Net) 阅读数 36
平台下的IKAnalyzer的C#移植版本。 支持Lucene.Nethttp://www.zgkw.cn/FORUMS/blogs/d... 博文 来自： iteye_17589的...

ne几个分词技术的比较 阅读数 3760
词器简单介绍 Lucene的分词技术很多，我下面介绍集中常用的分词技术。1）标准分词技术... 博文 来自： wittdong的博客

几个主要的Lucene中文分词器的比较 阅读数 174
介绍：paoding：Lucene中文分词“庖丁解牛”PaodingAnalysisimdict：imdict智能词典所采... 博文 来自： iteye_3678的...

ne--5.支持中文分词 阅读数 408
析器1.1. 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... 博文 来自： 村西头的俏寡妇

ne的各中文分词比较 阅读数 1135
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... 博文 来自： Think

lucene中文分词 阅读数 1362
是一个全文检索引擎工具包，貌似挺好用。某些时候我们需要在数据库全表扫描筛选数据时，如... 博文 来自： jiangzhongwe...

ne之中文分词器（IK-Ananlyzer）-yellowcong 阅读数 1176
全名为IKAnalyzer，是由java编写的中文分词工具包，目前在lucene以及solr中用的比较多，采... 博文 来自： yelllowcong的...

ne初探(二):中文分词,以及系统自带分词简单比较 阅读数 1460
学习我不得不承认这门技术是我目前接触的最有难度的一门技术,也许是因为我最近比较浮躁吧,... 博文 来自： 都市桃源

ne学习二：lucene分词器 阅读数 2747
的作用:在创建索引时会用到分词器，在使用字符串搜索时也会用到分词器，这两个地方要使用同... 博文 来自： 荒唐的程序猿

ne全文搜索之分词器：使用IK Analyzer中文分词器（修改IK Analyzer源码使... 阅读数 5594
单介绍下IKAnalyzerIKAnalyzer是linliangyi2007的作品，再此表示感谢，他的博客地址：http:/... 博文 来自： eguid

ne的建立索引，搜索，中文分词 阅读数 3595
是apache软件基金会4jakarta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... 博文 来自： young_so_nic...

ne6.5.0 下中文分词IKAnalyzer编译和使用 阅读数 4104
cene本省对中文分词有支持，不过支持的不好，其分词方式是机械的将中文词一个分成一个进行... 博文 来自： 暴走抹茶樊樊樊

ne 4.4.0中常用的几个分词器 阅读数 1万+
hitespaceAnalyzer以空格作为切词标准，不对语汇单元进行其他规范化处理。很明显这个实用... 博文 来自： ceclar123的专栏

几个流行的Lucene中文分词器对比 阅读数 445
：http://www.iteye.com/news/96371.基本介绍：paoding ：Lucene中文分词“庖丁解牛”P... 博文 来自： lkx94的专栏

一步跟我学习lucene（4）---lucene的中文分词器jcseg和IK Analyzer分词器... 阅读数 5351
要使用lucene中文分词器在lucene的开发过程中，我们常会遇到分词时中文识别的问题，lucene... 博文 来自： wuyinggui的...

ne 6.0下使用IK分词器 阅读数 7517
6.0使用IK分词器需要修改修改IKAnalyzer和IKTokenizer.使用时先新建一个MyIKTokenizer类,... 博文 来自： 1.01^365=37....

ne中文分词器Jceseg和IK Analyzer使用示例 05-09
文分词器Jceseg和IK Analyzer使用示例,lucene5可以使用，本人亲测成功，大家放心用，喜欢lucene的人大家关注我的博客 http://bl... 下载

ne基础（三）-- 中文分词及高亮显示 阅读数 3554
分词器及高亮分词器在lucene中我们按照分词方式把文档进行索引，不同的分词器索引的效果... 博文 来自： fun

检索lucene中文分词的一些总结 阅读数 1万+
索几乎是所有内容管理系统软件（CMS）必备的功能，在对公司的CMS产品的开发维护过程中... 博文 来自： 路卫杰的专栏

高版本Lucene的中文分词器（IK分词器）的DEMO 阅读数 372
前项目中需要用到Lucene.且需要中文分词，看了下IK分词器，但是IK分词器貌似只支持到lucen... 博文 来自： 诺浅的专栏

ne开源中文分词器 IKAnalyzer2.0.2 共享及源码发布 03-01
yzer2.0.2源代码 博文链接：https://linliangyi2007.iteye.com/blog/165287 下载

ne系列三：Lucene分词器详解、实现自己的一个分词器 阅读数 309
cene分词器详解1.Lucene-分词器API（1）org.apache.lucene.analysi.Analyzer分析器，分词... 博文 来自： 冷夜轩的博客

ne 中文分词器如何扩充中文词库啊
中文分词器如何扩充中文词库啊 求帮助？？？1534432371@qq.com这是我的邮箱 在帮着找找这个... 论坛

ne的中文分词器 阅读数 540
的中文分词器到现在还没有好的解决办法。下边介绍了两个lucene自己提供的分词器和一个java... 博文 来自： 墨竹

ne中文分词器(三) 阅读数 524
绍1.1 分词器（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... 博文 来自： pfnie的博客

ne+ikanalyzer实现中文同义词搜索 阅读数 4382
实现索引的创建与检索；ikanalyzer实现对中文的分词；光到这里已经能够实现中文的检索了，... 博文 来自： yax405的专栏

Lucene的案例开发：分词器介绍 阅读数 6729
ne创建索引的过程中，分词技术是一个十分重要的环节，介绍了7中比较常见的分词技术CJKAna... 博文 来自： 小鸡慢慢的专栏

ne之分词器效果测试 阅读数 59
果二、代码测试不同分词器，只需要将下面代码替换为需要测试的分词器Analyzeranalyzer=ne... 博文 来自： 绣花针

ne使用 阅读数 28
lt;!--ikanalyzer分词器--><dependency><groupId>... 博文 来自： fengyingsuixu...

ne 分词原理 阅读数 54
是一个高性能的java全文检索工具包，它使用的是倒排文件索引结构。该结构及相应的生成算法... 博文

ıe 分词器Analyzer 阅读数 1336
cene的analyisis包下的Standard包下的StandardAnalyzer()功能很强大,英文的处理能力同于St... 博文 来自： shendeguang...

验索之lucene的优化篇--分词器 阅读数 1764
索引库的基础上，加上中文分词器的，更好的支持中文的查询。引入jar包je-analysis-1.5.3.jar,... 博文 来自： suchy

ıe分词器分词 阅读数 586
jecom.essearch.core.analyzer;importjava.io.IOException;importjava.io.Reader;importjava.... 博文 来自： 云守护的专栏

ne-----查看分词结果 阅读数 1309
-----查看分词结果 博文 来自： hekewangzi

ne自带的分词器分词操作 阅读数 1203
自带的分词器分词操作：SimpleAnalyzerStopAnalyzerWhitespaceAnalyzerStandardAnalyz... 博文 来自： 墨竹

ne分词原理与方式 阅读数 1675
----------lucene的分词_分词器的原理讲解----------... 博文 来自： Jonney's house

ne中常用的几个分词器 阅读数 2494
页：http://blog.csdn.net/ceclar123/article/details/10150839一、WhitespaceAnalyzer以空... 博文 来自： 和尚敲出有节...

一步跟我学习lucene（3）---lucene的analysis相关和自定义分词器 阅读数 4572
分词相关总结和自定义分词器已经停止词词典的维护,自定义分词 博文 来自： wuyinggui的...

ne、solr中文分词器 10-02
认自带的分词器对中文支持并不好，所以对于中文索引的分词器，建议使用第三方开源的中文分词器 下载

ne中文分词器的比较 阅读数 352
介绍：paoding ：Lucene中文分词 “庖丁解牛” PaodingAnalysisimdict ：imdict智能词典所... 博文 来自： burpee的博客

ıe三---中文分词器 阅读数 372
z分析器1.1.1. Lucene自带中文分词器l StandardAnalyzer：单字分词：就是按照中文一个字一... 博文 来自： youfashion的...

ne的中文分词器IKAnalyzer 阅读数 1907
的中文分词器IKAnalyzer分词器对英文的支持是非常好的。 一般分词经过的流程： 1）... 博文 来自： 不了了之专栏

ıe的中文分词器 阅读数 939
的中文分词器lucene的中文分词器到现在还没有好的解决办法。下边介绍了两个lucene自己提供... 博文 来自： bubei的专栏

简单功能强大的excel工具类搞定excel导入导出工具类(一) 阅读数 4万+
EE项目导入导出Excel是最普通和实用功能,本工具类使用步骤简单,功能强大,只需要对实体类进行... 博文 来自： 李坤 大米时代 ...

检测工具face_recognition的安装与应用 阅读数 6万+
测工具face_recognition的安装与应用 博文 来自： roguesir的博客

R别，移植到android系统 阅读数 3万
槽，搞了1天半，终于弄好了。自己android开发是小白，之前一门心思想在jni目录下读取xml文... 博文 来自： Where there i...

学习----支持向量机（核函数） 阅读数 2022
扩展到非线性可分领域 博文 来自： 欢迎来到我的...

oid 合并生成分享图片（View截图） 阅读数 1万+
以前写过的自定义课表软件 ,Android 自定义View课程表表格 原生View截图合成分享的图片 看... 博文 来自： ShallCheek

esky分解法 阅读数 3万+
ky分解法又叫平方根法，是求解对称正定线性方程组最常用的方法之一。对于一般矩阵，为了消... 博文 来自： ACdreamer

OS SSH安装和配置 阅读数 7743
S SSH安装和配置 赞0 CentOS SSH 安装 配置 OpenSSH SSH 为 Secure Shell 的缩写，由 IET... 博文 来自： 耕耘——从菜...

J 数码是否有解 牛人总结 归并排序 阅读数 3940
的博客 先介绍八数码问题： 我们首先从经典的八数码问题入手，即对于八数码问题的任意一个... 博文 来自： hnust_xiehon...

JI - 一个简单的后台管理系统入门实例 阅读数 2万+
syUI 1.4.x 版本，默认default风格，异步加载页面，多Tab页展示，使用JSON文件模拟从后台... 博文 来自： 般若

凸包+点到平面距离+已知3点求平面方程 阅读数 3375
==========*\ | 3D凸包 | CA... 博文 来自： 南方公园

os 查看命令源码 阅读数 9万+
install yum-utils 设置源: [base-src] name=CentOS-5.4 - Base src - baseurl=http://vault.ce... 博文 来自： linux/unix

g boot实战(第五篇)配置源码解析 阅读数 1万+
面的文章都采用markdown编写的，但编辑图片上极其不方便，以后还是采用网页的形式。 上... 博文 来自： liaokailin的专栏

e 布隆过滤器BloomFilter介绍 阅读数 2万+
功能 提高随机读的性能 2、存储开销 bloom filter的数据存在StoreFile的meta中，一旦写入无... 博文 来自： opensure的专栏

ab并行编程方法 阅读数 9万+
一下matlab中的并行方法与技巧。分为以下几个板块： 1. 什么东西好并行？ 2. 怎么并行？ 3. p... 博文 来自： Rachel Zhang...

el文件导入数据库（POI+Excel+MySQL+jsp页面导入）第一次优化 阅读数 3万+
章是根据我的上篇博客，给出的改进版，由于时间有限，仅做了一个简单的优化。相关文章：将... 博文 来自： Lynn_Blog

图片服务器《二》-linux安装nginx 阅读数 4万+
个好东西，Nginx (engine x) 是一个高性能的HTTP和反向代理服务器，也是一个IMAP/POP3/... 博文 来自： maoyuanmin...

上安装Docker(非常简单的安装方法) 阅读数 22万+
校有空，大四出来实习几个月了，作为实习狗的我，被叫去研究Docker了，汗汗！ Docker的三... 博文 来自： 我走小路的博客

y/js实现一个网页同时调用多个倒计时(最新的) 阅读数 46万+
/js实现一个网页同时调用多个倒计时(最新的) 最近需要网页添加多个倒计时. 查阅网络,基本上都... 博文 来自： Websites

聊聊系统设计：有状态、无状态 阅读数 1万+
从线程安全的角度聊了聊系统设计要注意的事情，这次换个角度继续聊聊系统设计 这次主题围绕... 博文 来自： Runtime.class

pringBoot bean无法注入的问题（与文件包位置有关） 阅读数 19万+
景描述整个项目通过Maven构建，大致结构如下： 核心Spring框架一个module spring-boot-b... 博文 来自： 开发随笔

通分量及缩点tarjan算法解析 阅读数 59万+
分量： 简言之 就是找环（每条边只走一次，两两可达） 孤立的一个点也是一个连通分量 使用t... 博文 来自： 九野的博客

【Java虚拟机】之五：多态性实现机制——静态分派与动态分派 阅读数 3万+

李阿昀 关注

原创	粉丝	喜欢	评论
469	3582	1141	1043

等级： 博客 访问： 199万+

积分： 1万+ 排名： 722

勋章：

归档

热门文章

最新评论