2009年
3月14日

开源全文检索引擎Lucene本地化实践研究

报告人：吴鹏飞
单　位：石家庄学院图书馆
Email: wupengfei_2000@163.com

- Lucene
- 系统架构
- 中英文语言分析器ZH_CNAnalyzer
- 实验结果
- 应用案例
- 总结

Lucene

- Lucene作为一个非常优秀开源全文检索引擎，是 Apache软件基金会jakarta项目组的一个子项目，是一个用Java编写的开源全文索引与检索引擎工具包。
- 应用到很多Java项目中如web论坛系统Jive、邮件列表系统 Eyebrows、基于XML的web发布框架Cocoon、Java开发平台Eclipse、机构知识库DSpace等。
- 在开源社区中，对应不同的编程语言也已经有多个版本，如Lucene.Net、CLucene、dotLucene、Plucene、PyLucene、Lupy等 。

特点

- 提供了灵活的接口函数
- 分块增量索引和批量索引
- 数据源灵活多样
- 索引字段可定制
- 索引文件独立于具体平台
- 面向对象的系统架构

系统架构

- Lucene采用面向对象的系统架构，共13个包，表1中列出了其中核心的7个包及其功能说明。

表 1　Lucene 核心包及功能

核心包名	功能说明
org.apache.lucene.analysis	语言分析器：主要用于分词，支持中文主要是扩展此类
org.apache.lucene.document	索引存储时的文档结构管理，类似于关系型数据库的表结构
org.apache.lucene.index	索引管理：包括索引建立、更新、删除等
org.apache.lucene.queryParser	查询分析器：实现查询关键词间的运算
org.apache.lucene.search	检索管理：根据查询条件，检索得到结果
org.apache.lucene.store	数据存储管理：主要包括一些底层的 I/O 操作
org.apache.lucene.util	一些公用使用类

系统架构

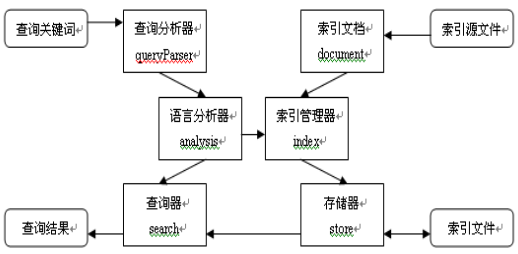


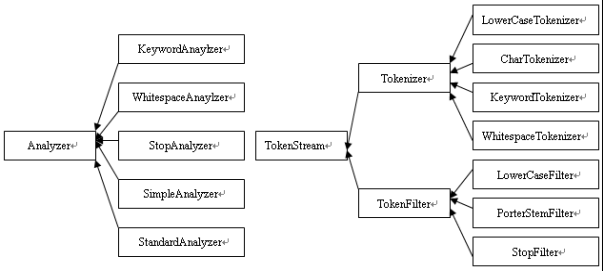
图 1　Lucene 索引与检索过程示意图

系统架构

- **Lucene**在索引与检索时都要调用语言分析器，语言分析器最主要的功能就是分词处理。文本分词技术在**Lucene**中起着非常重要的作用，如果**Lucene** 没有良好的中文分词功能，就不能对中文文本建立高效的索引以及检索。
- **Lucene**默认提供了**2**个比较通用的分析器**SimpleAnalyzer**和**StandardAnalyzer**，其支持中文分词，但是只能进行单字分词，功能十分有限。扩展语言分析器**CJKAnalyzer**支持中文双字分词，但是索引与检索效果不是很理想。

中英文语言分析器ZH_CNAnalyzer

- 语言分析器中主要包括**Analyzer**、**TokenStream**、**Tokenizer**和**TokenFilter**四个抽象基类及其继承类



中英文语言分析器ZH_CNAnalyzer

- **ZH_CNAnalyzer**类是语言分析器抽象类**Analyzer**的继承类，主要功能调用中英文分词类完成中英文分词，并按**Token**标准格式返回**Tokens**流。

```
public class ZH_CNAnalyzer extends Analyzer {
 public final static String[] STOP_WORDS ;// 中英文停用词表如"a"、"的"
 private Set stopWords;                    //停用词集合
 private static fenci fc;                  //定义中英文分词变量
 public ZH_CNAnalyzer(){
    fc=new fenci();                       //构造中英文分词对象
    fc.ReadDicwordToHash();               //读词典到哈希索引表
    stopWords = StopFilter.makeStopSet(STOP_WORDS); //设置停用词过滤
 }
 ......
 public TokenStream tokenStream(String fieldName, Reader reader) {
    String inputStr = readerTostring(reader);// Reader转换为String
    String resultStr;                         //定义分词结果变量
    resultStr = fc.Fenci(inputStr);       //中英文分词处理
    TokenStream result=new ZH_CNTokenizer(resultStr);//转换为Tokens流
    ......
    return result;                            //返回Tokens流
 }}
```

中英文语言分析器ZH_CNAnalyzer

- **ZH_CNTokenizer**类是**Tokenizer**抽象类的继承类，主要功能是按**Token**标准格式装配**Token**，具体包括词本身、起始位置、中英文词类型。

```
public class ZH_CNTokenizer extends Tokenizer {
    String [] words=null;                    //定义词数组变量
    public ZH_CNTokenizer(Reader in) throws Exception{
       input = in;
       words= ReadToArray(input);          //分词结果变换成数组格式
    }
    public Token next() throws IOException {
       Token token=null;
       String word=null;
       .....
       return new Token(word, start, end, tokenType); //返回标准的Token
    }
}
```

实验结果

No	Rank	Field	Text
42	1	<contents>	万
43	1	<contents>	三
44	1	<contents>	上
45	1	<contents>	下
46	1	<contents>	不
47	1	<contents>	与
48	1	<contents>	专
49	1	<contents>	世
50	1	<contents>	业
51	1	<contents>	东
52	1	<contents>	中
53	1	<contents>	为
54	1	<contents>	主
55	1	<contents>	之

图 3　单字分词索引结果

No	Rank	Field	Text
42	1	<contents>	一代
43	1	<contents>	一夜
44	1	<contents>	万篇
45	1	<contents>	三届
46	1	<contents>	上诊
47	1	<contents>	上学
48	1	<contents>	上招
49	1	<contents>	上海
50	1	<contents>	上申
51	1	<contents>	下一
52	1	<contents>	下周
53	1	<contents>	下载
54	1	<contents>	不包
55	1	<contents>	不易

图 4　双字分词索引结果

No	Rank	Field	Text
110	1	<contents>	体系
111	1	<contents>	作品
112	1	<contents>	作文
113	1	<contents>	促进
114	1	<contents>	保送生
115	1	<contents>	信息
116	1	<contents>	信息化
117	1	<contents>	信息网
118	1	<contents>	债务
119	1	<contents>	假冒
120	1	<contents>	假期
121	1	<contents>	做好
122	1	<contents>	元旦
123	1	<contents>	光华
124	1	<contents>	免试

图 5　ZH_CNAnalyzer 分词索引结果

应用案例

- 具体调用中英文语言分析器**ZH_CNAnalyzer**对本地磁盘文件建立索引与检索的实例。

```
File   indexDir = new File("F:\\数字图书馆\\index"); //存储索引目录
File   sdir  = new File("F:\\数字图书馆"); //索引文件源目录
File[] sfiles  = sdir.listFiles(); //存储文件数组
Analyzer    analyzer = new ZH_CNAnalyzer();//定义中英文语言分析器
IndexWriter indexWriter = new IndexWriter(indexDir, analyzer,true);
//定义索引器，调用ZH_CNAnalyzer
for(int i = 0; i < sfiles.length; i++){ //循环访问文件
  if(sfiles[i].isFile() && sfiles[i].getName().endsWith(".txt")){
    Document document = new Document(); //定义文档对象
    Reader txtReader = new FileReader(sfiles[i]);
    document.add(new Field("path", sfiles [i].getCanonicalPath(),
        Field.Store.YES,Field.Index.NO)); //文件路径存储不索引
    document.add(new Field("contents",readerTostring(txtReader),
        Field.Store.YES,Field.Index.TOKENIZED));//文件内容存储并索引
    indexWriter.addDocument(document);//添加文档写入索引
```

应用案例

```
Searcher searcher=new IndexSearcher(indexPath);//指向索引目录的搜索器
Analyzer analyzer = new ZH_CNAnalyzer();
QueryParser parsecontent=new QueryParser("contents", analyzer);//
查询解析器：使用与索引同样的语言分析器ZH_CNAnalyzer
Query query =parsecontent.parse(queryString);//解析关键字
Hits hits = searcher.search(query);//搜索结果使用Hits存储
for (int i=0; i<hits.length(); i++) {
   System.out.println(hits.doc(i).get("path")+";Score:" +
      hits.score(i)); //通过hits检索path字段文件路径数据和查询的匹配度
);
```

总结

- 中英文语言分析器**ZH_CNAnalyzer**，使**Lucene**具有很好的中文处理能力，能够对中英文混合文档完成多字分词索引存储；
- 应用到数字图书馆中来实现对数字资源的全文检索；
- 应用到资源采集系统如**Nutch**中。