1. 信息检索模型
2. 信息检索概述
3. 分词算法
4. 英文分词

处理流程为：输入文本->词汇分割->词汇过滤（去除停留词）->词干提取（形态还原）->大写转换为小写->结果输出

1. 中文分词
   1. 词典匹配分词法
      1. 正向最大匹配
      2. 逆向最大匹配
      3. 最少切分
   2. 语义理解分词法
   3. 词频统计分词法
2. 倒排索引

简单的解释就是： 正排索引：文档中有哪些单词

倒排索引：单词都处于哪些文档

1. 布尔检索模型

运用布尔运算符连接各个检索词

布尔运算符：

AND （或\*）：逻辑与，如A AND B，同时包含A和B的集合

OR（或+）：逻辑或，如A OR B,包含A和B之一的集合

NOT(或-)：逻辑非，如A NOT B，包含A但不包含B的集合

运算符优先级：NOT>AND>OR, 如中国NOT日本AND歌曲OR小说，索引结果为：名字包含中国但是不包含日本的歌曲或小说

也可以用（）来任意结合

1. tf-idf权重计算

词频-逆文档频率，用以计算词项对于一个文档集或一个语料库中的一份文件的重要程度。词项的重要性与在一个文档中出现的频率成正比，但与在一个文档集中出现的频率成反比。

（1） tf为词项频率，词频的标准化有以下两种计算方法

tf = 单词在文档中出现的次数/文档的总次数

tf = sqrt(单词在文档中出现的次数) #Lucene中采用的方法

（2） df为文档频率，代表文档集中包含某个词的所有文档数目，df通常比较大，把它映射到一个较小的取值范围，用逆文档频率(idf)来表示：

idf = log(文档集中的文档数/(包含某个词的文档数+1)) = log(N/(df+1))

1. tf-idf为词项的权重，计算公式如下：

tf-idf = tf(词频)\*idf(逆文档频率)

通过tf-idf可以把文档表示成n维词权重向量

1. 向量空间模型

利用余弦定理来判单查询关键字与文档之间的相似度

1. 概率检索模型
   1. 贝叶斯决策理论
   2. 二项独立模型
   3. Okapi BM25模型
   4. BM25F模型
2. Lucene开发入门
3. Lucene分词详解

lucene中分词主要以靠Analyzer类解析实现

Tokenizer类处理单个字符组成的字符流，读取Reader对象中的数据，处理后转换成词汇单元

TokenFilter:完成文本过滤器的功能，使用过程中必须注意不同过滤器的使用顺序

创建索引的时候需要分词器，进行索引查询的时候也需要分词器，这两个地方要使用同一种分词器，lucene提供了一下分词器的方法：

|  |  |
| --- | --- |
| StopAnalyzer(停用词分词器) | 能过滤词汇中的特定字符产和词汇，并且完成大写转小写的功能 |
| StandardAnalyzer(标准分词器) | 根据空格和符号来完成分词，还可以完成数字，字母，E-mail地址，ip地址以及中文字符的分析处理，还可以支持过滤词表，用来代替StopAnalyzer能够实现的过滤功能 |
| WhitespaceAnalyzer(空格分词) | 使用空格最为间隔符的词汇分割器，不完成单词过滤和小写字符转换，不需要分词词库支持 |
| SimpleAnalyzer(简单分词) | 具备基本西文字符词汇分析的分词器，处理词汇单元时，以非字母字符作为分隔符好。不能用作词汇的过滤，只能进行词汇 的分析和分割。 |
| CJKAnalyzer(二分法分词) | 对中文进行分词，同时使用StopFilter过滤器完成过滤功能，实现中文的多元切分和停用词过滤 |
| KeywordAnalyzer(关键词分词) | 把整个输入作为一个单独词汇单元，方便特殊类型的文本进行索引和检索。针对邮政编码、地址等文本信息使用关键词分词器进行索引项建立非常方便 |

1. Lucene索引详解
2. Lucene字段类型

|  |  |
| --- | --- |
| 字段类型 | 说明 |
| TextField | 会把该字段的内容索引并词条化，但是不保存词向量 |
| StringField | 只对该字段的内容索引，但是并不词条化，也不保存词向量 |
| IntPoint | 适合索引值为int类型的字段 |
| LongPoint | 适合所应之为long类型的字段 |
| FloatPoint | 适合所应之为float类型的字段 |
| DoublePoint | 适合所应之为double类型的字段 |
| SortedDocValuesField | 适合索引字段值为文本内容并且需要按值进行排序的字段 |
| SortedSetDocValuesField | 适合索引字段值为文本内容并且需要按值进行分组，聚合等操作的字段 |
| NumericDocValuseField | 存储单个数值类型的DoccValues字段 |
| SortedNumericDocValuseField | 存储数值类型的有序数组列表的DOcValues字段 |
| StoredField | 适合索引只需要保存字段值不进行其他操作的字段 |

1. lucene索引文档的创建
   * + 1. lucene索引文档的创建需要依靠一个IndexWriter对象，创建IndexWriter需要提供两个参数，一个是IndexWriterConfig对象，设置创建索引使用哪种分词器，另一个是索引的路径。IndexWriterConfig对象的setOpenMode()方法可以设置索引的打开方式
       2. IndexWriter对象的addDocument()方法用于添加文档，该方法的参数为Document对象
       3. FieldType对象的setIndexOptions()方法可以设定域（字段）的索引选型，参数如下：
          1. IndexOptions.DOCS:只索引文档，词项频率和位移信息不存在
          2. IndexOptions.DOCS\_AND\_FREQS：只索引文档和词项频率，位移信息不保存
          3. IndexOptions.DOCS\_AND\_FREQS\_AND\_POSITIONS：索引文档，词项频率和位移信息
          4. IndexOptions.DOCS\_AND\_FREQS\_AND\_POSITIONS\_AND\_OFFSETS：索引文档，词项频率，位移信息和偏移量
          5. IndexOptions.NONE：不索引
       4. 为了准确地获取位置信息以及一些偏移量就需要在创建索引地时候进行纪律，如果信息不准确那么可能在搜索地时候会发生错位，在索引地时候可以使用FieldType对象提供地方法设置对应增量和位移信息
          1. setStored(boolean value) : 参数默认值为false，设置为true存储字段值
          2. setTokenized(boolean value) : 参数设置为true，会使用配置地分词器对字段值进行词条化
          3. setStoreTermVectors( Boolean value) : 参数为true,保存词向量
          4. setStoreTermVectorPositions(boolean value) : 参数为true,保存词项的词向量中的位移信息。
          5. setStoreTermVectorOffsets( Boolean value) : 参数为true,保存词项在词向量中的偏移信息