搜索引擎总结

这个学期通过对蔡老师《搜索引擎》课程的学习以及自己动手实际编写了一个“典故搜索引擎”。使我对搜索引擎有了一个更深的认识，特别是在做搜索引擎过程中对于各个搜索引擎技术的解决，既使我深入了解了搜索引擎的细节技术，也使我对这些技术的实现有了充分的掌握。下面是我在做“典故搜索引擎”过程的总结。

1. 搜索引擎概念

要想做一个搜索引擎，我们就必须知道什么是搜索引擎。

搜索引擎是指根据一定的策略、运用特定的计算机程序从互联网上搜集信息，在对信息进行组织和处理后，为用户提供检索服务，将用户检索相关的信息展示给用户的系统。

1. 搜索引擎体系结构

由以上的搜索引擎概念，并结合自己对搜索引擎的认识，得出以下搜索引擎体系结构图：

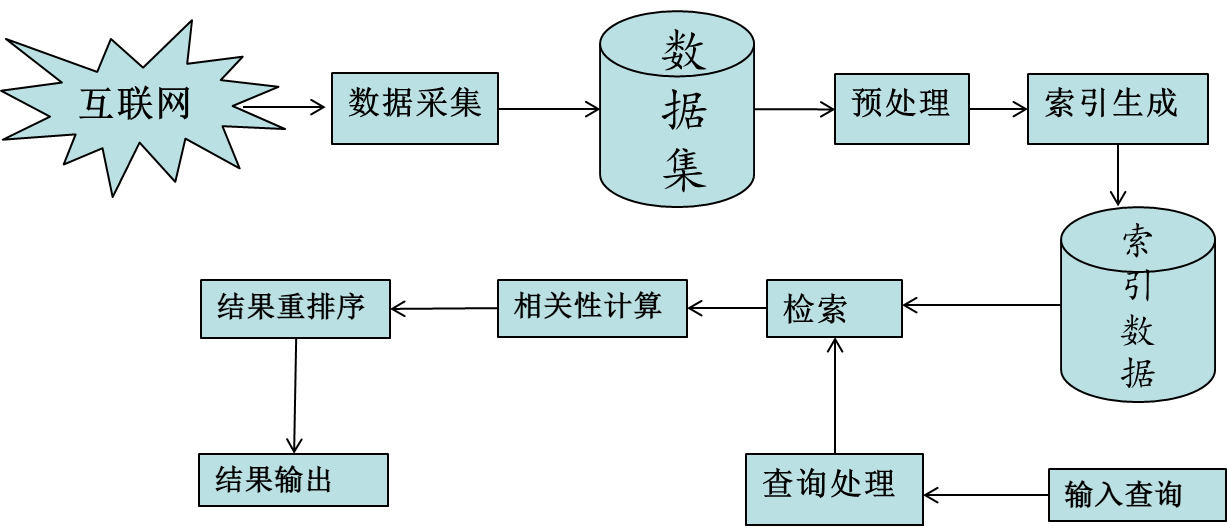


图1 搜索引擎体系结构

1. 搜索引擎的主要任务

从搜索引擎的概念和体系结构图中，我得出做一个搜索引擎需要的三大步骤：数据搜集、数据处理、数据查询。而其中数据处理又可以分为网络爬虫和网页解析两步；数据处理分为数据预处理、建立索引；数据查询分为数据检索和用户接口两个部分。

因此我得出建立搜索引擎的6个步骤：

1. 设计和实现网络爬虫并爬取网络数据；
2. 解析网页数据，得到自己需要的数据；
3. 对数据进行预处理
4. 建立索引
5. 处理查询检索
6. 设计友善的用户接口
7. 数据搜集——爬虫

现在进入构建搜索引擎的第一步设计和实现网络爬虫并爬取需要的网络数据。数据采集是搜索引擎的数据来源其主体主要是网页数据搜集

作为网页数据搜集的重要组成部分——爬虫，本系统采用了开源软件Heritrix。

选择Heritrix的原因有两点：

（1）Heritrix是由Java语言编写，而本实例的其他部分也都采用了Java语言编写，因此采用Heritrix方便功能的实现。

（2）Heritrix是一款开源的爬虫，并且扩展性很好，可以方便的修改源代码以适应我们的需求。

**Heritrix**的工作流程是一个循环，具体流程是：

　　1 在预定的URL中选择一个。

　　2 从选择的URL的网址下载远程文件

　　3 分析，归档下载到的内容

　　4 从分析到的内容里面选择感兴趣的URL。加入预定队列。

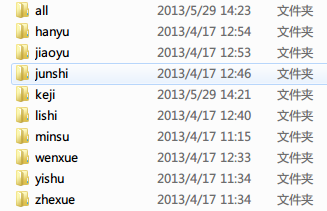
　　5 标记已经处理过的URL

在建立好Heritrix工程并对它的网络爬虫的下载功能进行相应的扩展之后，就可以使用Heritrix进行对于指定的网络数据资源的下载。



图2 Heritrix对指定网络资源下载

在Heritrix上对所需要的功能进行配置后进行下载任务，下载到的数据资源如下：



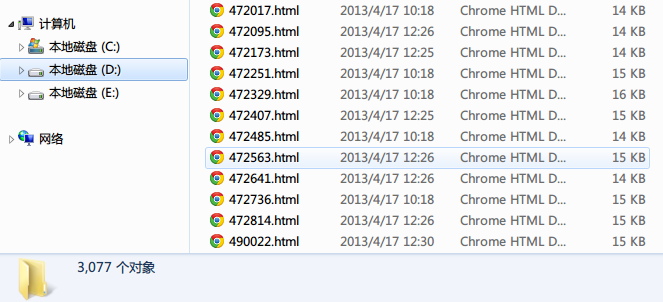


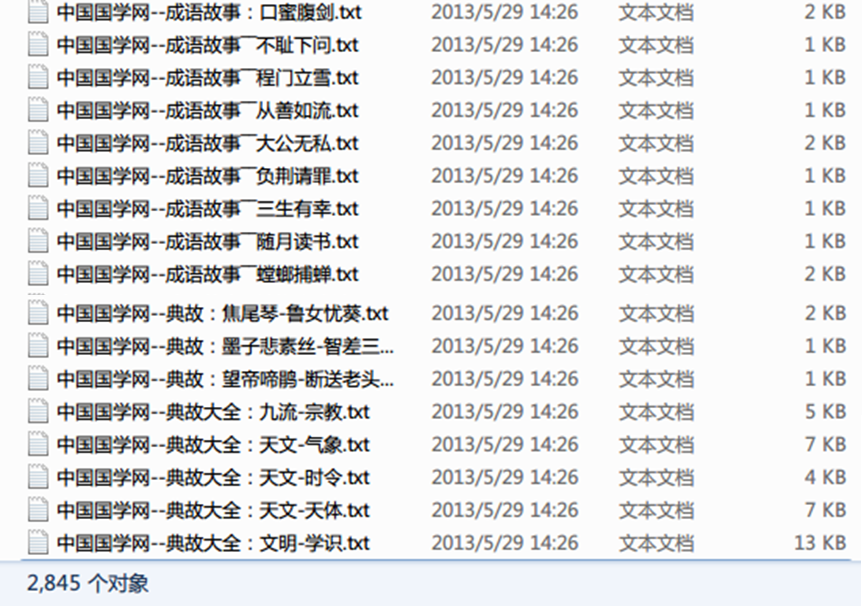
图3 Heritrix下载的网络资源

1. 数据搜集——网页解析

在使用Heritrix下载到了需要的网络数据资源后，重要的一个步骤就是对这些网络数据资源进行解析。原始的网络数据一般都是非结构化的如Html、pdf、doc、ppt等，其中以Html文档占绝大多数，而Html文档中包含很多无用的标签，以及广告链接等，如何从Html中解析出我们需要的信息，是我们需要思考和设计的。一般有两种方法可以对Html进行解析：1、使用正则表达式对网页数据进行解析；2、使用Htmlparser对Html进行解析。对于第一种方法，正则表达式的编写是一件比较复杂的事情，需要花更多的时间在正则表达式的修改和测试上，但当正则表达式书写正确时，解析的正确度比较高。对于第二种方法，由于它是开源的Html解析工具，已经封装了Html解析的各种方法，掌握起来非常容易，而且解析效果也非常高。因此我选择的是使用Htmlparser来解析网页数据。得到需要的信息内容，去除冗余信息。我们得到包含：名称、摘要、类型、内容、图片URL、网页URL、更新时间这些需要的信息。



图4 Htmlparser核心代码



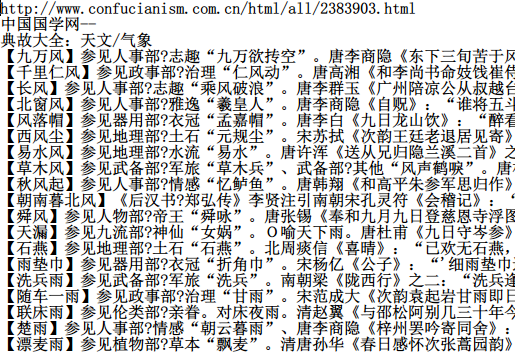


图5 Htmlparser解析后的数据

1. 数据处理

数据处理部分是搜索引擎中的核心组成部分，它包括对数据的预处理、索引文件的生成及建立索引数据库。

数据处理部分主要应用Lucene开源工具。

Lucene作为一个全文检索引擎，其具有如下突出的优点：

（1）索引文件格式独立于应用平台。

（2）在传统全文检索引擎的倒排索引的基础上，实现了分块索引，能够 针对新的文件建立小文件索引，提升索引速度。然后通过与原有索 引的合并，达到优化的目的。

（3）优秀的面向对象的系统架构，使得对于Lucene扩展的学习难度降低，方便扩充新功能。

（4）设计了独立于语言和文件格式的文本分析接口，索引器通过接受Token流完成索引文件的创立，用户扩展新的语言和文件格式，只需要实现文本分析的接口。

（5）已经默认实现了一套强大的查询引擎，用户无需自己编写代码即使系统可获得强大的查询能力，Lucene的查询实现中默认实现了布尔 操作、模糊查询（Fuzzy Search）、分组查询等。

开源,可扩展能力强,有各种语言版本,适合各种平台。

6.1、数据预处理

在进行建立索引之前需要对数据进行一些预处理工作：

（1）将文档的全角标点转成半角标点

（2）将大文档切分成多个小文档

（3）预处理源文件的统一接口

（4）为索引的建立预准备字典

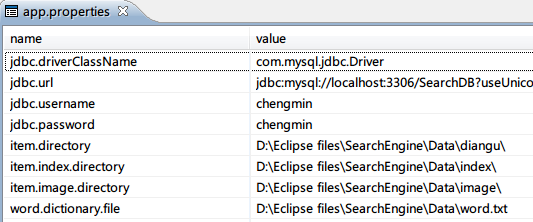


图6 为文件路径及数据库建立统一接口

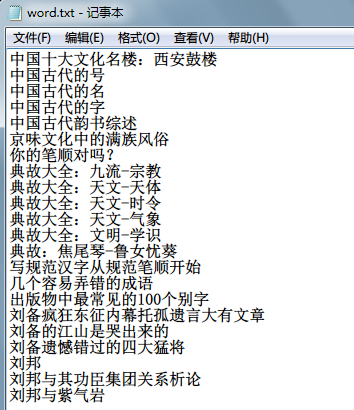


图7 预处理后的数据字典

6.2、Lucene建立索引

索引是搜索引擎最重要的一部分Lucene通过对IndexWriter的addDocument接口，将构建好的Document加入索引

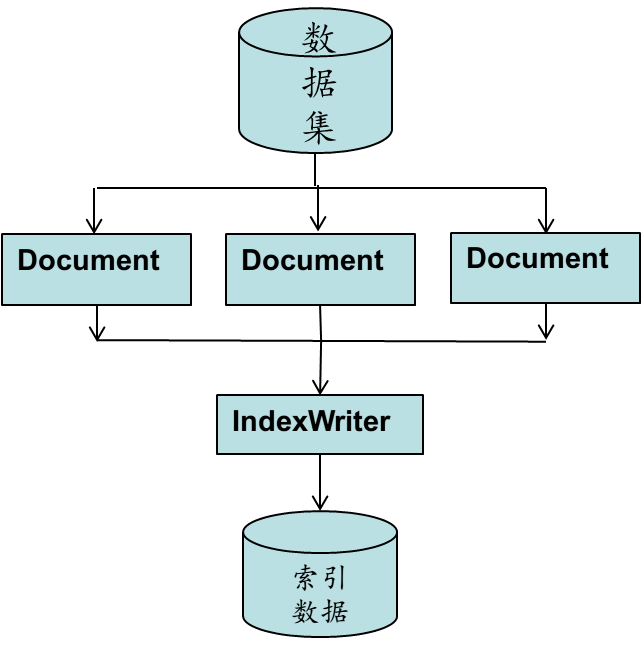


图8 lucene建立索引

在lucene的API中，IndexWriter的主要职责是向索引中添加文档，它提供了建立索引的主要外部接口。

在DocumentWriter中完成了向索引添加文档的工作后，lucene会进一步对索引的一些情况进行判断，并做进一步的合并索引的操作。

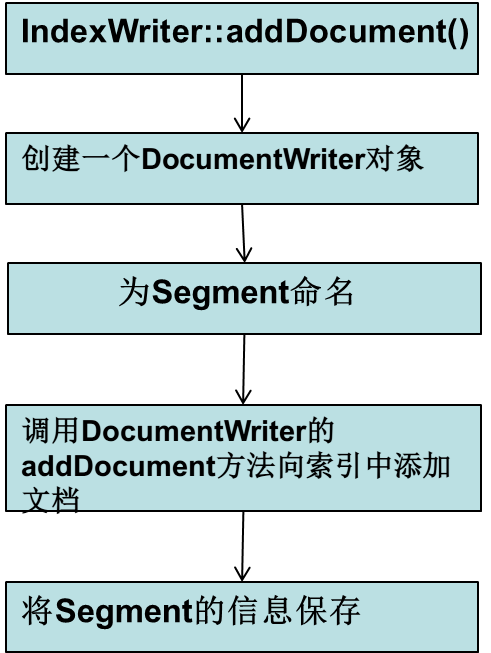


图9 lucene建立索引详细过程

6.3、Lucene分词

在lucene建立索引的时候，写入索引并能够被用户检索的是一个个词条。只有通过分词才可能让信息检索系统理解用户的检索请求，进而为其搜索相关的内容。

lucene自带分词器Analyzer。

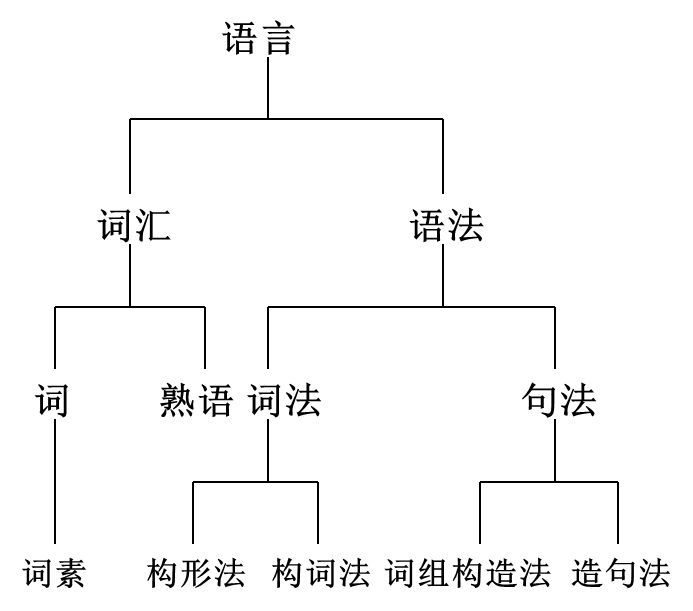


图10 分词模型

1. 数据查询

数据查询是搜索引擎功能的体现，也是搜索引擎系统与用户交互的接口。对于搜索引擎开发者来说，数据处理阶段是搜索引擎的核心，它体现了整个搜索引擎的效率。而对于使用搜索引擎的使用者来说，其最关心的问题是数据查询，及查询结果的显示。

在数据查询阶段，首先由搜索引擎使用者输入查询，有搜索引擎接收查询并在索引数据库中进行检索，并将检索出来的数据进行相关性的计算和结果重排，最后将数据结果以合理的方式输出给用户。

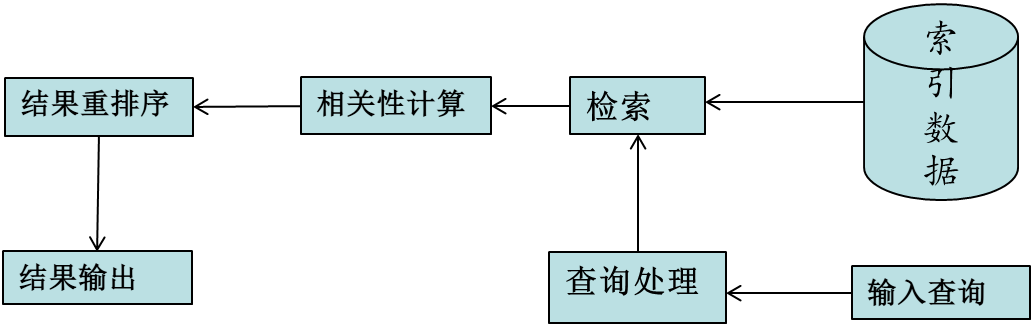
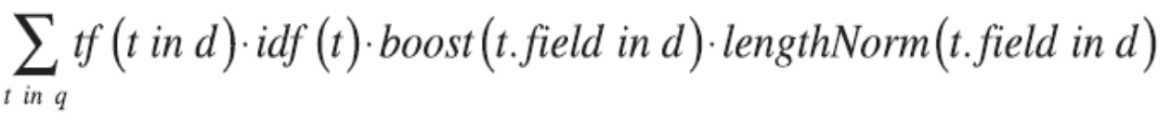


图11 数据查询

7.1、相关性计算与结果重排

按照lucene的文档得分进行排序



* + **tf(t in d)词条t在文档d中出现的词频**
  + **idf( t )词条t在文档中的倒排词频**
  + **boost(t.field in d)在索引过程中设置的字段参数**
  + **lengthNorm(t.field in d)字段的标准化值，表明在字段中存储了多少词条，这个数值是在索引过程中计算出来的，并且也存储在索引中**
  + **coord(q, d)协调因子，它的计算是基于文档d中所包含的所有可供查询的词条数量**
  + **queryNorm(q)在给出每个查询条目的方差和后，计算某查询的标准化值**

**可以使用Document的setBoost方法来改变文档的boost因子。这样我们就得到了自己的结果排序。**

1. 搜索引擎工程结构

在完成了以上的建立搜索引擎步骤和技术后，一个整体的搜索引擎系统也就建立完成。在建立搜索引擎过程中，采用了模块化开发方法，把整个工程分为了数据库、索引、配置、过滤、处理、搜索、服务、测试等八大模块，增加了开发的简明性和效率

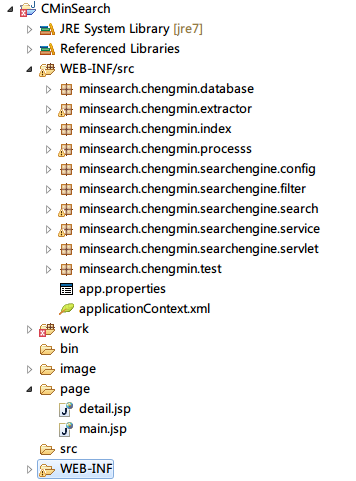
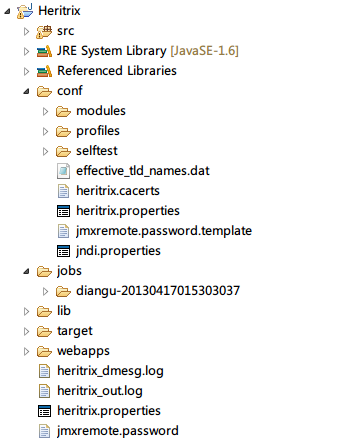


图12搜索引擎工程结构

1. 用户接口

对于一个优秀的搜索引擎来说，强大的搜索功能和搜索效率是其必不可少的，但一个友好的用户接口同样占到很大的分量，一个友好的用户接口决定了搜索引擎能不能受到用户的欢迎。

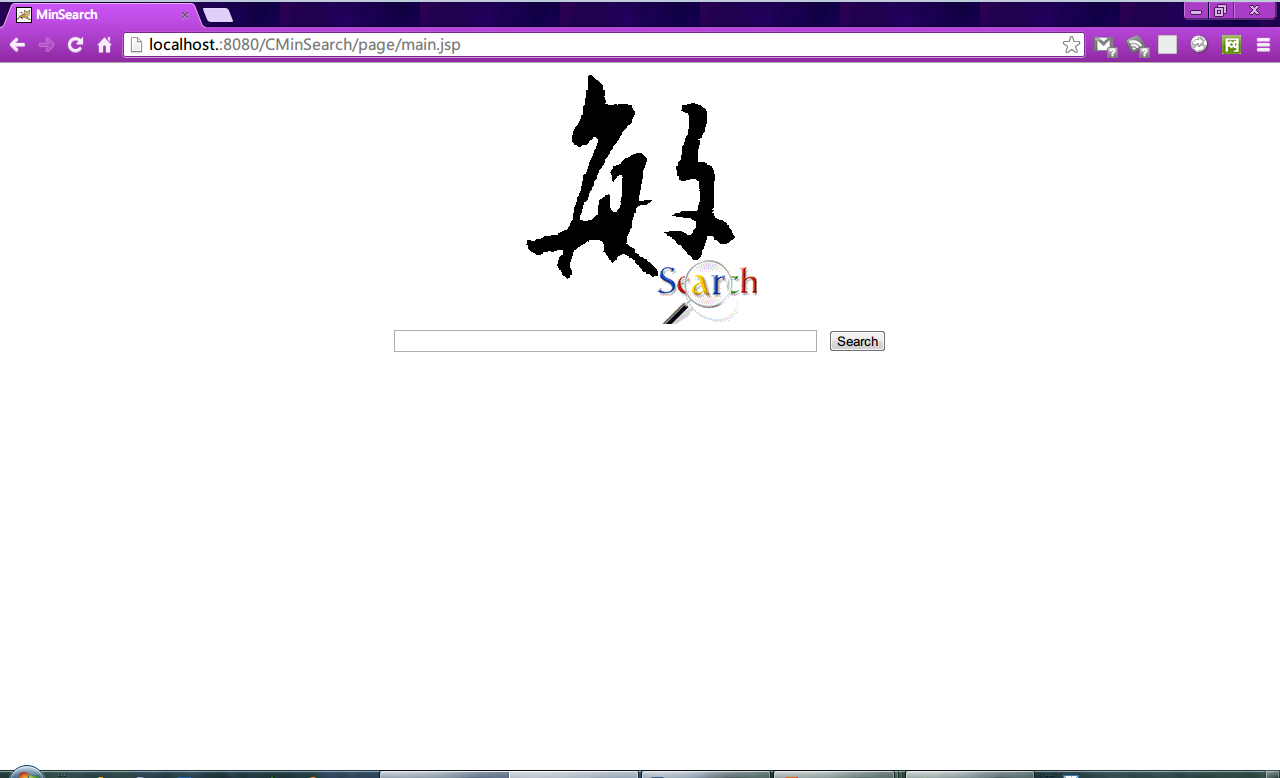


图13搜索引擎用户接口

1. 搜索效果展示



图13搜索引擎搜索效果展示