



附超值案例

开发自己的 搜索引擎

Lucene + Heritrix (第2版)

© 邱哲 符滔滔 王学松 编著

畅销书升级 原书是国内第一本讲解搜索引擎开发的畅销书

超值 提供了价值上万元的大型数码产品搜索引擎开发案例，可直接应用于项目

版本最新 采用了最新的Heritrix-1.14.0版、HTMLParser 1.6.0版、DWR 2.0.5版

实践性强 用案例的方式讲解，便于读者实践

注重原理讲解 提供了结构框图和流程图，讲解搜索引擎的原理



人民邮电出版社
POSTS & TELECOM PRESS



开发自己的

搜索引擎

Lucene + Heritrix (第2版)

© 邱哲 符滔滔 王学松 编著

人民邮电出版社
北京

图书在版编目 (C I P) 数据

开发自己的搜索引擎 : Lucene+Heritrix / 邱哲, 符滔滔, 王学松编著. — 2版. — 北京 : 人民邮电出版社, 2010. 1

ISBN 978-7-115-21529-1

I. ①开… II. ①邱… ②符… ③王… III. ①计算机网络—程序设计 IV. ①TP393.09

中国版本图书馆CIP数据核字(2009)第182347号

内 容 提 要

本书是一本介绍搜索引擎开发的书籍,通过本书,读者可以独立构建一个企业级的搜索引擎网站。本书讲解了搜索引擎与信息检索基础, Lucene 入门实例,索引的建立,使用 Lucene 来搜索,排序,分析器,对 Word、Excel 和 PDF 格式文档的解析, Compass 搜索引擎框架, Lucene 分布式,爬虫 Heritrix, HTMLParser, DWR 等内容。最后综合这些技术,构建了一个典型的垂直搜索系统,具有很强的商业实用价值。

本书是一本使用 Lucene 和 Heritrix 来讲解搜索引擎构建的书,通过对 API 和源代码的分析,力求使读者在应用的基础上,能够深入其核心,自行扩展和开发相应组件,发挥想象力,开发出更具有创意的搜索引擎产品。

本书适合 Java 程序员和从事计算机软件开发的编程人员阅读,同时也可以作为搜索引擎爱好者的入门书籍。

开发自己的搜索引擎——Lucene+Heritrix (第2版)

- ◆ 编 著 邱 哲 符滔滔 王学松
责任编辑 屈艳莲
- ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京艺辉印刷有限公司印刷
- ◆ 开本: 800×1000 1/16
印张: 36
字数: 773 千字 2010 年 1 月第 2 版
印数: 1—3 500 册 2010 年 1 月北京第 1 次印刷

ISBN 978-7-115-21529-1

定价: 69.00 元 (附光盘)

读者服务热线: (010)67132692 印装质量热线: (010)67129223

反盗版热线: (010)67171154

第2版 前言

2007年初,我们编写了本书第1版,该书曾经连续数周占据互动出版网计算机类畅销书排行榜首。转眼间两年过去了,我收集了很多读者的建议和问题,针对书中的细节进行了调整,推出了本书的第2版。

第2版主要在以下方面进行了改进。

(1) 对第1版中语言进行了优化,以使行文更加流畅,便于阅读,同时对一些表达模糊的地方进行了改写。

(2) 针对读者提出的一些问题,进行了勘误。

(3) 对书中的大部分 Visio 图片进行了重绘,看起来更美观。

(4) 对书中涉及的软件进行全面梳理,对版本进行了更新,主要包括以下两部分。

- Lucene 采用了 2.0 稳定版,正文中的范例都是使用它来实现的。目前 Lucene 已经发布到了 2.4 版,但是这个版本还没有大规模的商业化应用,存在很多不稳定因素,因此笔者只在附录中介绍了其核心功能。

- 网络爬虫采用了 Heritrix 1.14.0 版本,包括后面的大案例都是使用这个版本重新实现的。

(5) 去除了第9章中已经不再使用的 Google 的 searchAPI 部分。

(6) 对于第12章,升级了内核代码版本,使用 Heritrix 1.14.0 版本,增加了网络爬虫 Heritrix 代码工程导入和配置的详细步骤。修改了具体的代码,解决由于来源网站内容变更造成的部分代码无法执行的问题。

(7) 对于第13章,增加了对网页内容分析的概述和基本说明,便于读者更方便地理解相关内容。使用 HTMLParser 1.60,适应网站代码的修改,变动了其中的正则分析代码和网页解析代码。

(8) 对于第15章和第16章,升级核心代码为 DWR 2.0.5 版本。针对技术手段,从搜索引擎交互方式角度进行了分析和介绍。增加了代码部分的图例说明。

本书由邱哲、符滔滔、王学松统筹编写,同时参与编写的还有王石、熊英、付京周、袁福庆、张杰、赵显琼、卜庆玲、常利、冯曼菲、匡妍娜、雷成健、李小波、刘浩然、刘会神、王晓悦、马震、齐志华、韩延峰、舒军、孙大林、孙佳楠、王辉、王沛等人,在此一并表示感谢。

编者

2009年10月

前 言

背景

搜索,这两个字无疑是当今互联网业界最为流行的字眼之一。在 Baidu 上输入“搜索引擎”这个关键字,可以找到 3000 多万篇的网页。在 Google 上查找时,可以查到 750 万个网页。不是 Google 的网页少,如果用“search engine”做关键字查找时,在 Google 中可以查找到 3 亿篇以上的网页!

再来做个有趣的实验。在 www.china-pub.com 中输入“搜索引擎”这个关键词后,只有可怜的七本书被查找了出来。

从上大学开始,我就知道, www.china-pub.com 应该是国内最大的计算机网上书店了。可是,为什么一个在 Google 中可以查找到 3 亿网页的关键字,在国内最大的计算机网上书店中只能找到可怜的、与之相关的七本书?

300000000 网页 VS 7 本书?

本书特点

由于目前市面上从技术层面介绍搜索引擎的书并不多,即使有,也大多停留在理论阶段,而非搜索引擎的开发过程。因此,可以说本书是一本详细介绍搜索引擎开发过程的图书。

(1) 采用的是 Lucene 2.0。以前大家用的 1.4.3 版本,而 Lucene 2.0 重写了 API,内部的实现方法也有了很大优化。本书的代码都是在 2.0 版本下调试通过的,这样可以帮助读者了解 Lucene 的更多新功能。

(2) 本书配有一个完整的搜索引擎案例。这个案例有很强的实用价值,只需稍加修改,就能应用于实际项目。

(3) 着重解决开发人员头痛的问题。本书的目的是指导项目实践,因此没有罗列各个 API 的用法,而是对常见的开发问题进行深入探讨,比如本书的第 7 章,是专门为解决“Word、Excel 和 PDF 文件如何解析”这个问题而设置的。

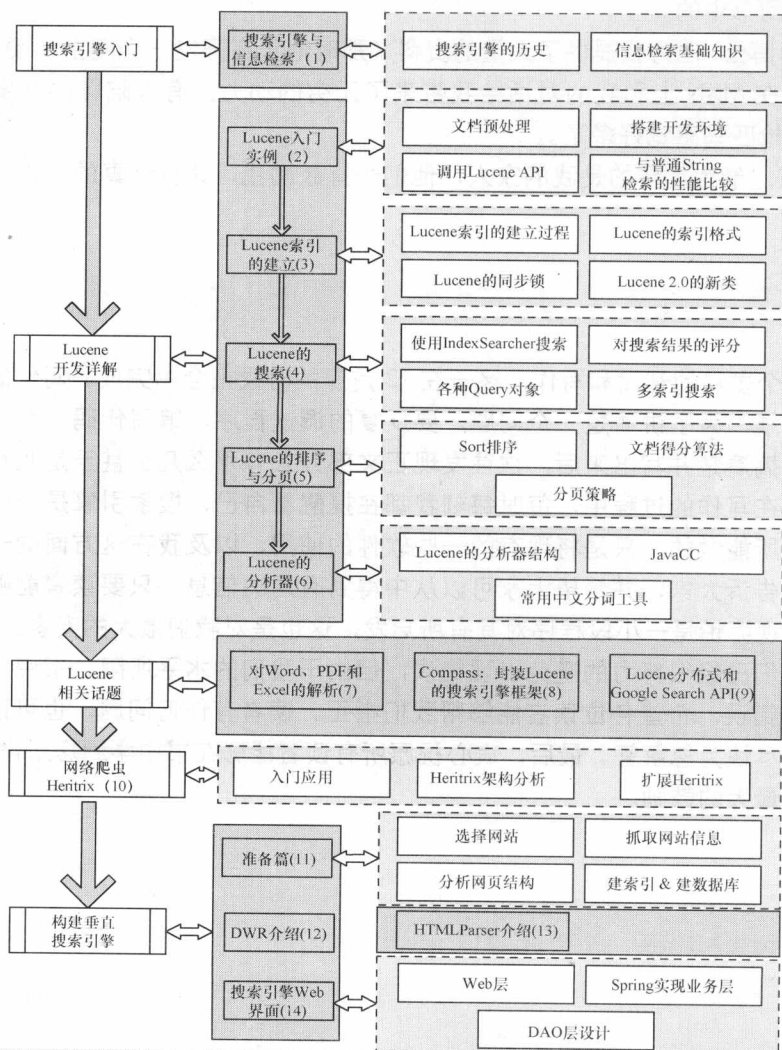
(4) 内容新颖、实用。本书介绍了 Compass、Heritrix、DWR 和 HTMLParser 等内容。在搜索引擎开发的过程中,这些均为相当重要且实用的技术,笔者经过自身实践将其展现给读者,希望能让读者在学习 Lucene 的同时开拓视野。

售后服务

我们为本书开通了专用的 BLOG，网址是 <http://lucenebook.spaces.live.com/>，读者可以直接同我们交流，共同学习和提高。另外，我们还为本书提供了专门的联系邮箱，luceneheritrix@163.com。读者可以随时同我们联系。

本书的内容

本书内容非常丰富，细节可以参考目录。在这里，笔者给出全书的结构图，让读者有一个总体的认识，其中深色背景，并且后面有 (1)、(2) 字样的，表示相应的章节。



感谢

在写这段前言的时候，ZZQ 正在忙着校对我们的稿件，要向他表示感谢，因为他是我见过的最为认真负责的审稿人。同时，这本书的写作过程中，还要感谢以下几个人。

- 吴萌野：可以说，如果没有他的指导，我不可能了解 Heritrix 的使用，它的技术水平我真是服了，他在一个月内踏平 Lucene 和 Heritrix 两座大山，开发出一个比价购物的搜索引擎。

- 何进，他是一位热心的读者，也是一个 Lucene 的爱好者，Compass 的内容便是他建议加入到书中的。

- 杜华博士，他为我提供了大量的资料（那时他正在开发一个爬虫），更重要的是，每天深夜他在 MSN 上和我的对话给我带来了无穷的动力。有人陪伴的开发过程，要比一个人单枪匹马感觉好多了。

- 当然，最为重要的是我的家人，他们一直鼓励我，让我认真的工作，是我坚强的后盾。

结束语

经过 6 个多月的开发和写作，终于在 12 月底的时候把全书完成。写作的过程是艰苦的，有时候，为了讲清楚一个问题，要反复的调试程序，编写代码。而当花了大量精力把问题搞清楚并写出来后，往往发现正文部分只有那么几页甚至是几行，但是我想说的是，在写作的过程中，每时每刻我都在提醒着自己，搜索引擎是一门博大精深的学科，我所能做的，只是将现有的一些软件的使用，以及我在这方面的一些微薄经验拿出来，告诉大家，并希望大家可以从中得到有用的信息。只要读者能够在阅读的过程中，发现哪怕是一小段程序对其有所启发，这也是对我们最大的安慰。

虽然我们已经很努力的避免出现错误，但由于我们的水平所限，书中不可避免的会出现一些错误，希望各位读者能够帮我们指正。读者有任何问题，也可以反馈给我们，我们会尽快为您解答。最后，衷心祝愿所有读者能够在书中学到您所需要的知识，这是对我们最大的鼓励。

目 录

第1章 搜索引擎与信息检索	1	2.3.1 准备工作	22
1.1 搜索引擎的历史	1	2.3.2 创建工程并引入 Lucene 的 JAR 包	24
1.1.1 萌芽: Archie、Gopher	1	2.3.3 运行文档预处理类	31
1.1.2 起步: Robot (网络机器人) 的出现与 Spider (网络爬虫) ...	3	2.3.4 创建处理文档的索引类: IndexProcessor	32
1.1.3 发展: Excite、Galaxy、 Yahoo 等	4	2.3.5 创建检索索引的搜索类	34
1.1.4 繁荣: Infoseek、AltaVista、 Google 和 Baidu	6	2.4 运行效果	38
1.2 信息检索系统的基本知识	9	2.5 小结	41
1.2.1 信息检索系统	9	第3章 索引的建立	42
1.2.2 信息检索的过程	11	3.1 Document 逻辑文件	42
1.2.3 传统查找的优点和不足	12	3.1.1 Lucene 的 Document	42
1.2.4 使用索引提高检索速度	12	3.1.2 为 Document 添加多种 Field ...	43
1.2.5 倒排索引	13	3.1.3 Document 的内部实现	45
1.2.6 评价信息检索系统的标准	14	3.2 Field 的内部实现	46
1.3 Lucene 的简介	14	3.2.1 Field 包含的类	47
1.4 小结	15	3.2.2 Field 类的构造方法	48
第2章 Lucene 入门实例	16	3.3 Lucene 的索引工具 IndexWriter ...	49
2.1 实例介绍	16	3.3.1 IndexWriter 的初始化	50
2.1.1 实例说明	16	3.3.2 向索引添加文档	52
2.1.2 开发过程	16	3.3.3 限制每个 Field 中的词条的 数量	53
2.2 准备工作	17	3.4 Lucene 索引过程详解	54
2.2.1 将文档的全角标点转成半角 标点	17	3.4.1 Lucene 索引建立过程概览	54
2.2.2 将大文档切分成多个小文档 ...	20	3.4.2 使用 addDocument 方法向 索引添加文档	55
2.2.3 预处理源文件的统一接口	21	3.4.3 DocumentWriter 的 addDocument 方法	57
2.3 创建 Eclipse 工程	22		

3.4.4	文档的倒排	62	4.1.2	IndexSearcher 的最简单使用	89
3.4.5	对 postingTable 进行排序	66	4.1.3	IndexSearcher 的多种 search 方法	90
3.4.6	将 Posting 信息写入索引	68	4.2	Hits 类详解	92
3.5	索引文件格式	68	4.2.1	Hits 类的公有接口	92
3.5.1	索引的 segment	69	4.2.2	效率分析	93
3.5.2	.fnm 格式	69	4.2.3	Hits 内部的缓存	95
3.5.3	.fdx 与 .fdt 格式	70	4.2.4	Hits 类的工作原理	98
3.5.4	.tii 与 .tis 格式	71	4.3	对搜索结果的评分	98
3.5.5	deletable 格式	71	4.3.1	文档与词条的向量空间	98
3.5.6	复合索引格式 .cfs	71	4.3.2	Lucene 的文档得分算法	99
3.6	索引过程的调优	72	4.4	构建各种 Lucene 内建的 Query 对象	103
3.6.1	合并因子 mergeFactor	72	4.4.1	toString: 查看原子查询	103
3.6.2	maxMergeDocs	73	4.4.2	查询重写与权重	103
3.6.3	minMergeDocs	73	4.4.3	TermQuery 词条搜索	104
3.7	索引的合并与索引的优化	74	4.4.4	BooleanQuery 布尔搜索	105
3.7.1	FSDirectory 与 RAMDirectory	74	4.4.5	RangeQuery 范围搜索	113
3.7.2	使用 IndexWriter 来合并索引	75	4.4.6	PrefixQuery 前缀搜索	117
3.7.3	索引的优化	76	4.4.7	PhraseQuery: 短语搜索	119
3.8	从索引中删除文档	78	4.4.8	MultiPhraseQuery: 多短语 搜索	123
3.8.1	索引的读取工具 Index- Reader	78	4.4.9	FuzzyQuery 模糊搜索	128
3.8.2	使用文档 ID 号来删除特定文档	81	4.4.10	WildcardQuery 通配符搜索	131
3.8.3	使用 Field 信息来删除批量 文档	84	4.4.11	SpanQuery 跨度搜索	132
3.9	Lucene 的同步问题	85	4.5	第三方提供的 Query 对象: RegexQuery	140
3.9.1	为什么要进行同步以及 Lucene 的同步法则	85	4.6	通过 QueryParser 转换用户 关键字	142
3.9.2	commit.lock 与 write.lock	85	4.6.1	词条的定义	143
3.10	Lucene 2.0 的新类: IndexModifier 类	86	4.6.2	QueryParser 初始化	143
3.11	小结	87	4.6.3	改变 QueryParser 默认的布尔 逻辑	144
第4章	Lucene 搜索	88	4.6.4	短语和 QueryParser	145
4.1	使用 IndexSearcher 进行搜索	88	4.6.5	FuzzyQuery 和 QueryParser	147
4.1.1	初始化 IndexSearcher	88			

4.6.6	通配符与 QueryParser	147
4.6.7	查找指定的 Field	148
4.6.8	RangeQuery 与 QueryParser	151
4.6.9	QueryParser 和 SpanQuery	152
4.7	多 Field 搜索与多索引搜索	153
4.7.1	多域搜索 MultiFieldQuery-Parser	153
4.7.2	MultiSearcher 在多个索引上搜索	155
4.7.3	ParallelMultiSearcher: 多线程搜索	158
4.7.4	Searchable 和 RMI	161
4.8	小结	162

第5章 排序、过滤和分页 163

5.1	相关度排序	163
5.1.1	使用 Score 进行自然排序	163
5.1.2	Searcher 的 explain 方法	165
5.1.3	通过改变 boost 值来改变文档的得分	166
5.2	使用 Sort 来排序	170
5.2.1	Sort 简介	170
5.2.2	SortField	171
5.2.3	按文档得分进行排序	172
5.2.4	按文档的内部 ID 号来排序	175
5.2.5	按一个或多个 Field 来排序	175
5.2.6	改变 SortField 中的 Locale 信息	182
5.3	搜索的过滤器	183
5.3.1	过滤器的基本结构	183
5.3.2	一个简单的 Filter: 建立索引	184
5.3.3	一个简单的 Filter: 打印索引文档信息	186
5.3.4	一个简单的 Filter: 安全级别与过滤器代码	187

5.3.5	一个简单的 Filter: 在搜索时应用过滤器	188
5.3.6	一个简单的 Filter: 总结	190
5.3.7	按范围过滤 RangeFilter	190
5.3.8	在结果中查询 QueryFilter	194
5.3.9	缓存结果: Caching-WrapperFilter	197
5.4	翻页问题	198
5.4.1	依赖于 session 的翻页	198
5.4.2	多次查询	198
5.4.3	缓存 + 多次查询	199
5.4.4	缓存 + 多次查询 + 数据库	199
5.5	小结	200

第6章 Lucene 的分析器 201

6.1	分析	201
6.1.1	分词	201
6.1.2	Lucene 的分析器结构	202
6.1.3	Lucene 的分析器实现	204
6.2	Lucene 与 JavaCC	205
6.2.1	JavaCC 简介	205
6.2.2	JavaCC 为 Lucene 提供的分析器脚本	206
6.2.3	Lucene 的标准分析器	210
6.2.4	标准过滤器: Standard-Filter	211
6.2.5	大小写转换器: Lower-CaseFilter	212
6.2.6	忽略词过滤器: StopFilter	213
6.3	分析器的进阶	213
6.3.1	再看 StandardAnalyzer 中的管道过滤器结构	214
6.3.2	长度过滤器: LengthFilter	214
6.3.3	PerFieldAnalyzerWrapper	215

6.3.4 其他	215	8.1.2 Compass 的代码片断	250
6.4 对中文的分析	216	8.2 Compass 的初始配置	252
6.4.1 现有的中文分词方式简介	216	8.2.1 Compass 的配置文件	252
6.4.2 中科院的分词软件和 JE 分词	218	8.2.2 将索引存放于内存中	253
6.5 小结	224	8.2.3 使用 JDBC 来存储索引	253
第7章 Word、Excel 和 PDF 的处理	225	8.2.4 使用连接池来存储索引	254
7.1 使用 PDFBox 处理 PDF 文档	225	8.2.5 加载 compass.cfg.xml 文件	255
7.1.1 PDFBox 的下载	225	8.3 域模型的配置	255
7.1.2 在 Eclipse 中配置	226	8.3.1 实体代码	255
7.1.3 使用 PDFBox 解析 PDF 内容	227	8.3.2 实体关系	261
7.1.4 运行效果	228	8.3.3 实体 Book 的配置文件	262
7.1.5 与 Lucene 的集成	230	8.3.4 通用元数据定义文件 (.cmd.xml)	263
7.2 使用 xpdf 来处理中文 PDF 文档	232	8.3.5 Author 和 Article 的配置 文件	267
7.2.1 xpdf 的下载	232	8.4 使用 Compass 来建立索引	269
7.2.2 配置	232	8.4.1 索引代码	269
7.2.3 提取中文	233	8.4.2 对象关系图和运行结果	271
7.2.4 运行效果	236	8.5 使用 Compass 来搜索	272
7.3 使用 POI 来处理 Excel 和 Word 文件格式	237	8.5.1 使用 find() 方法搜索	272
7.3.1 对 Excel 的处理类	237	8.5.2 CompassHits 类型	273
7.3.2 ExcelReader 的运行效果	241	8.5.3 CompassHit 类型	274
7.3.3 POI 中 Excel 文件 Cell 的 类型	243	8.5.4 使用 Lucene 语法来查找	275
7.3.4 对 Word 的处理类	245	8.6 配置 Analyzer 和 Optimizer	276
7.4 使用 Jacob 来处理 Word 文档	247	8.7 小结	277
7.4.1 Jacob 的下载	247	第9章 Lucene 分布式	278
7.4.2 在 Eclipse 中配置	247	9.1 Lucene 与分布式	278
7.5 小结	249	9.1.1 什么是 GFS	278
第8章 Compass: 封装了 Lucene 的 框架	250	9.1.2 为 Lucene 提供分布式的 几点设想	279
8.1 Compass 简介	250	9.2 小结	281
8.1.1 Compass 的下载	250	第10章 无比强大的网络爬虫 Heritrix	282
		10.1 Heritrix 的使用入门	282

- 10.1.1 下载和运行 Heritrix 282
- 10.1.2 在 Eclipse 里配置 heritrix 的
开发环境 285
- 10.1.3 创建一个新的抓取任务 290
- 10.1.4 设置抓取时的处理链 292
- 10.1.5 设置运行时的参数 295
- 10.1.6 运行抓取任务 297
- 10.1.7 Heritrix 的镜像存储结构 302
- 10.1.8 终止抓取或终止 Heritrix
的运行 303
- 10.2 Heritrix 的架构 304
 - 10.2.1 抓取任务 CrawlOrder 304
 - 10.2.2 中央控制器 CrawlController 305
 - 10.2.3 Frontier 链接制造工厂 308
 - 10.2.4 用 Berkeley DB 实现的
BdbFrontier 313
 - 10.2.5 Heritrix 的多线程
ToeThread 和 ToePool 316
 - 10.2.6 处理链和 Processor 319
- 10.3 扩展和定制 Heritrix 322
 - 10.3.1 向 Heritrix 中添加自己的
Extractor 323
 - 10.3.2 定制 Queue-assignment-
policy 两个问题 327
 - 10.3.3 定制 Queue-assignment-policy
继承 QueueAssignmentPolicy
类 328
 - 10.3.4 扩展 FrontierScheduler
来抓取特定的内容 329
 - 10.3.5 在 Prefetcher 中取消
robots.txt 的限制 330
- 10.4 小结 331
- 第 11 章 搜索引擎综合实例：
准备篇 332**
 - 11.1 数码产品垂直搜索引擎
实例简介 332
 - 11.1.1 垂直搜索引擎实现流程 332
 - 11.1.2 数码垂直搜索引擎搜索
功能 333
 - 11.1.3 信息来源网站的选择方法 333
 - 11.1.4 太平洋电脑网和网易
手机频道 334
 - 11.2 准备 Eclipse 的 Web 开发环境 335
 - 11.2.1 准备 Eclipse 的 Web 插件
环境 335
 - 11.2.2 在 Eclipse 中配置插件 336
 - 11.3 准备垂直搜索引擎工程 337
 - 11.3.1 建立搜索引擎 Eclipse 工程 338
 - 11.3.2 设置搜索引擎工程上下文
信息 339
 - 11.3.3 设定源代码存放和输出
路径 340
 - 11.3.4 添加自定义的 Java 代码 341
 - 11.3.5 添加工程中引用的 Jar 包 343
 - 11.3.6 创建工程 JSP 页面文件 345
 - 11.3.7 构造完成的工程整体结构 347
 - 11.4 搜索引擎配置信息管理及相关类 349
 - 11.4.1 工程配置信息管理 349
 - 11.4.2 系统属性配置文件 350
 - 11.4.3 配置文件管理封装类 350
 - 11.5 小结 352
- 第 12 章 搜索引擎综合实例：下载篇 353**
 - 12.1 数码产品网络爬虫 353
 - 12.1.1 垂直搜索引擎网络爬虫设计 353
 - 12.1.2 来源网站内容与链接分析 354
 - 12.2 数码产品信息来源列表准备 356
 - 12.2.1 太平洋电脑网待抓取内容
页面分析 356

- 12.2.2 太平洋电脑网带抓取内容
代码分析 359
- 12.2.3 太平洋电脑网手机品牌
清单分析 362
- 12.3 Eclipse 中定制数码产品
Heritrix 爬虫 367
- 12.3.1 数码产品 Heritrix 爬虫的
功能 367
- 12.3.2 Eclipse 中导入编译 Heritrix
工程 368
- 12.3.3 Eclipse 中运行 Heritrix 工程 370
- 12.4 抓取 pconline 网页的定制
扩展类 371
- 12.4.1 抓取 pconline 网页的
Frontier 扩展 371
- 12.4.2 执行 pconline 手机网页
抓取任务 373
- 12.5 抓取网易手机频道的定制
扩展类 375
- 12.5.1 网易手机频道结构分析 375
- 12.5.2 设计网易抓取的 Extractor
扩展 378
- 12.5.3 设计网易抓取的 Frontier
扩展 381
- 12.5.4 执行网易手机频道网页
抓取任务 382
- 12.6 小结 383
- 第 13 章 使用正则表达式与 HTMLParser
分析网页 384**
- 13.1 网页内容分析方法概述 384
- 13.1.1 网页 HTML 的基本知识 384
- 13.1.2 JDK 正则表达式简介 385
- 13.1.3 HTMLParser 开源库介绍 387
- 13.2 正则表达式精确提取网页内容 388
- 13.2.1 正则表达式 java.util.regex
使用 388
- 13.2.2 正则表达式提取 tom 星座
内容实例 390
- 13.2.3 正则表达式提取 pconline
手机品牌列表 396
- 13.3 HTMLParser 高效提取网页
内容 398
- 13.3.1 HTMLParser 使用准备 398
- 13.3.2 Lexer 模式功能及实现 399
- 13.3.3 HTMLParser 功能及实现 404
- 13.3.4 HTMLParser 解析星座
网页实例 410
- 13.4 数码产品网页内容解析系统 413
- 13.4.1 产品详细信息文件格式 413
- 13.4.2 解析产品网页信息的基类
Extractor 414
- 13.5 pconline 手机产品网页内容
解析 418
- 13.5.1 pconline 手机产品页面
Extractor 解析器 418
- 13.5.2 pconline 产品信息解析
测试函数 421
- 13.5.3 pconline 产品信息解析
代码执行结果 422
- 13.6 网易手机频道产品内容解析 425
- 13.6.1 网易手机频道产品信息的
Extractor 解析器 425
- 13.6.2 网易手机频道的产品信息
运行测试效果 428
- 13.7 小结 429
- 第 14 章 网页内容存储与索引 430**
- 14.1 构建产品检索名称信息词库 430
- 14.1.1 产品名称词汇选择 430

- 14.1.2 产品名称词库提取代码 431
- 14.1.3 产品名称词库提取结果 433
- 14.2 手机产品数据库与文件索引
 - 结构 434
 - 14.2.1 手机产品的存储方法 434
 - 14.2.2 手机产品信息 Product 类 435
 - 14.2.3 产品信息数据库存储结构 437
 - 14.2.4 产品信息 Lucene 索引结构 438
- 14.3 产品信息数据库存储与处理 439
 - 14.3.1 数据库创建与准备 439
 - 14.3.2 Java 数据库基本操作 440
 - 14.3.3 数码产品数据库记录操作 441
- 14.4 产品信息文件存储与 Lucene 索引 443
 - 14.4.1 数码产品 Lucene 索引操作设计 443
 - 14.4.2 数码产品具体索引操作代码 445
- 14.5 产品信息综合处理与运行 446
 - 14.5.1 调用数据库处理类和索引处理类 446
 - 14.5.2 数码产品数据处理类运行 452
- 14.6 小结 454
- 第 15 章 搜索引擎综合实例：交互篇** 455
 - 15.1 DWR 的技术介绍 455
 - 15.1.1 Ajax 与 DWR 简介 455
 - 15.1.2 Ajax 与传统模式搜索架构 456
 - 15.2 DWR 安装与配置 457
 - 15.2.1 DWR 的下载与安装 457
 - 15.2.2 创建工程结构 458
 - 15.2.3 配置 web.xml 内容 460
 - 15.2.4 建立配置 dwr.xml 内容 461
 - 15.3 DWR 入门与实例演示 461
 - 15.3.1 简单 Ajax 页面代码 461
 - 15.3.2 运行效果与对比 464
 - 15.3.3 DWR 与直接使用 XMLHttpRequest 对象的比较 468
 - 15.3.4 在 DWR 中操纵自定义的对象 470
 - 15.3.5 查看 DWR 的输出日志 477
 - 15.4 dwr.xml 的配置进阶 477
 - 15.4.1 dwr.xml 的标准结构 478
 - 15.4.2 <init> 标签与 DWR 自带的 converter 和 creator 479
 - 15.4.3 <allow> 标签 483
 - 15.4.4 <signature> 标签 484
 - 15.4.5 转换器 converter 485
 - 15.5 使用 DWR 工具库 util.js 488
 - 15.5.1 页面中调用 util.js 489
 - 15.5.2 使用 useLoadingMessage() 方法显示提示图标 490
 - 15.5.3 DWRUtil.setValue() 和 DWRUtil.getValue() 495
 - 15.5.4 DWRUtil.getValues 和 DWRUtil.setValues 498
 - 15.5.5 DWRUtil.addOptions 和 DWRUtil.removeAll-Options 503
 - 15.5.6 DWRUtil.addRow 和 DWRUtil.removeAll-Rows 508
 - 15.5.7 DWRUtil.toDescriptive-String 方法 515
 - 15.6 小结 516
- 第 16 章 搜索引擎综合实例：Web 篇** 517
 - 16.1 Web 配置文件 517

16.1.1	配置文件及其作用	517	16.5.2	数码搜索手机产品图片的 显示	542
16.1.2	Spring 配置文件	518	16.5.3	手机产品详细信息页面 detail.jsp	543
16.1.3	DWR 配置文件	519	16.6	实例中的问题与功能扩展	546
16.1.4	web.xml 配置文件	520	16.7	小结	548
16.2	各种搜索相关 Bean 类	521	附录 Lucene 2.4 更新内容	549	
16.2.1	产品 SearchResult 结果 记录类	522	F1	IndexWriter 的构造函数	549
16.2.2	产品 SearchResults 结果 集合类	524	F2	IndexWriter 的 init 方法	550
16.2.3	产品 SearchRequest 检索 请求类	526	F3	IndexWriter 中的 flush、commit 和 close	552
16.3	数据库访问 SearchResultDAO 类实现	527	F4	Lucene 2.4 中的 Segment	553
16.3.1	数码库访问类接口定义	527	F5	IndexCommit 和 IndexDeletion- Policy	555
16.3.2	数码库访问类实现	528	F6	IndexWriter 中的 add- Document	558
16.4	Lucene 索引检索 SearchService 类实现	530	F7	DocumentsWriter 类的 add- Document 方法	559
16.4.1	索引检索类接口定义	530	F8	DocumentsWriter 的索引链	562
16.4.2	索引检索类实现	531			
16.5	前台 Web 页面设计	536			
16.5.1	数码垂直搜索主页面 main.jsp	536			



第1章 搜索引擎与信息检索

Google 的巨大成功让整个世界都把眼光投入到搜索引擎这个领域中。一夜间, 各种各样的搜索服务席卷而来, 从最初的 Google、Yahoo 到现今的 Baidu、MSN、中搜、Sogou 等, 搜索引擎的品牌越来越多, 服务也越来越丰富。同时, 伴随着 Web2.0 的疯狂普及, 网络信息的膨胀速度成指数急速增长, 各种各样的网站都需要为其加入检索功能, 以满足用户的需要。另外, 在企业级应用的市场上, 全文信息检索的需求也一直在增加, 各种文档处理、内容管理软件都需要加入全文检索的功能。

在这样的背景下, 搜索引擎的技术迅速发展。各种讨论搜索的文章、杂志、论文铺天盖地, 论坛和博客上的帖子也是层出不穷。一时间, 搜索技术成为最为热门的技术之一。

不过, 搜索引擎技术并非是一种大众技术, 从其出现开始, 就一直是一种高门槛的技术, 它的后台包括学术领域的众多先进思想和设计, 其涉及的学科包括自然语言处理、人工智能、离散数学、排列组合、编译原理等。因此设计一个性能良好, 并且实用性强的搜索引擎并非易事。

本书不研究上述多种学科与搜索引擎的关联理论, 但是读者了解和掌握搜索引擎技术的方方面面, 会对阅读有很大的帮助。因此, 在本章中, 将带领读者了解一下搜索引擎和信息检索的基础知识、发展历史、现今状况等内容。

1.1 搜索引擎的历史

在互联网发展的最初阶段, 网站的数量相对较少, 信息查找比较容易。随着互联网爆炸性地发展, 用户很难找到所需的资料。这时, 搜索引擎的需求就出现了, 一些为满足大众信息检索需求的专业搜索网站也就应运而生。

1.1.1 萌芽: Archie、Gopher

1. Archie

事实上, 搜索引擎的诞生追溯到 1990 年, 在加拿大蒙特利尔 (Montreal) 的麦吉尔大学, 一个学生制作了一个自动索引互联网上匿名 FTP 网站文件的程序。

这个能够自动索引互联网上匿名 FTP 网站文件的程序, 被他们称为 Archie。Archie 是

互联网上用来查找文档的自动搜索服务工具，这些文档的标题必须满足特定条件。

通常，为了从匿名 FTP 服务器上下载一个文件，必须知道这个文件的所在地，同时必须知道这个匿名 FTP 服务器的地址，及文件所在的目录名。Archie 可以帮助用户在遍及全世界的千余个 FTP 服务器中寻找文件。Archie Server 又被称作文档查询服务器。用户只要给出所要查找文件的全名或部分名字，文档查询服务器就会指出在哪些 FTP 服务器上的哪个路径下存放着这样的文件。使用 Archie 进行查询的前提是，输入要查找的文件名或部分文件名，知道某个或几个 Archie 服务器的地址。

如今，提供 Archie 服务的网站已经很少了，笔者在 Google 上查找了一下，链接到了一个波兰的网站，仍在提供着 Archie 服务，如图 1-1 所示。有兴趣的读者可以上去一看。

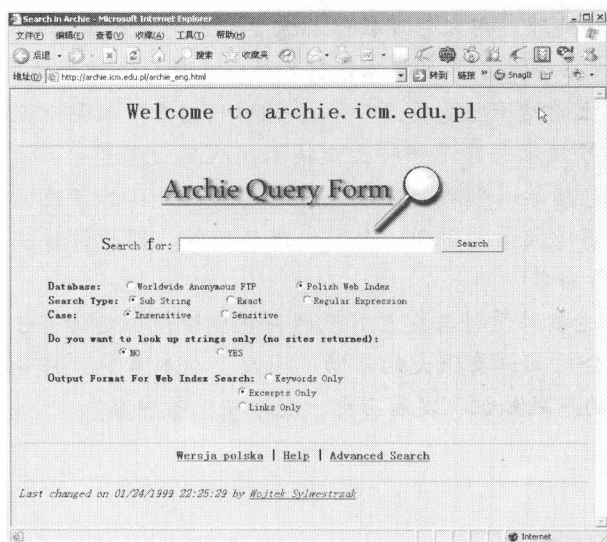


图 1-1 一个 Archie 网址

从概念上讲，Archie 的工作十分简单。每隔一段时间，一个特殊的程序连到每一个已知的匿名 FTP 主机，然后下载所有公共文件的完整目录表。这些表存储于 Internet Archives Database (Internet 档案数据库) 中。当用户要求 Archie 检索一个文件时，所要进行的工作就是对该数据库进行检索。

2. Gopher

受其启发，美国明尼苏达大学的一个学生 Mark McCahill，于 1991 年发明了一种叫 Gopher 的搜索协议。Gopher 的命名来自于这所学校的吉祥物。这种协议与 Archie 最大的不同是，Archie 仅能够索引网络上的文件，而 Gopher 却可以对网页也进行索引。同时，另外两个程序 Veronica 和 Jughead 用来对以 Gopher 格式进行索引的文件进行检索。Veronica (Very Easy Rodent-Oriented Net-wide Index to Computerized Archives) 是指非常方