Broadview®
www.broadview.com.cn

解密搜索引擎技术实战
Lucene & Java 精华版

第3版
全新升级

罗刚　等编著

包含本书全部实例代码
可用于工程实践
DVD

中国工信出版集团
電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

解密搜索引擎技术实战

Lucene & Java 精华版

第3版

罗刚　等编著

電子工業出版社
Publishing House of Electronics Industry
北京•BEIJING

内 容 简 介

本书是猎兔搜索开发团队的软件研发和教学实践的经验汇总。本书总结搜索引擎相关理论与实际解决方案，并给出了 Java 实现，其中利用了流行的开源项目 Lucene 和 Solr，而且还包括原创的实现。

本书主要包括总体介绍部分、爬虫部分、自然语言处理部分、全文检索部分以及相关案例分析。爬虫部分介绍了网页遍历方法和如何实现增量抓取，并介绍了从网页等各种格式的文档中提取主要内容的方法。自然语言处理部分从统计机器学习的原理出发，包括了中文分词与词性标注的理论与实现及在搜索引擎中的应用等细节，同时对文档排重、文本分类、自动聚类、句法分析树、拼写检查等自然语言处理领域的经典问题进行了深入浅出的介绍，并总结了实现方法。在全文检索部分，结合 Lucene 介绍了搜索引擎的原理与进展。用简单的例子介绍了 Lucene 的最新应用方法，包括完整的搜索实现过程：从完成索引到搜索用户界面的实现。此外还进一步介绍了实现准实时搜索的方法，展示了 Solr 的用法以及实现分布式搜索服务集群的方法。最后介绍了在地理信息系统领域和户外活动搜索领域的应用。

图书在版编目（CIP）数据

解密搜索引擎技术实战：Lucene＆Java 精华版 / 罗刚编著. —3 版. —北京：电子工业出版社，2016.4
ISBN 978-7-121-28111-2

Ⅰ. ①解… Ⅱ. ①罗… Ⅲ. ①互联网络－情报检索 Ⅳ. ①G354.4

中国版本图书馆 CIP 数据核字（2016）第 022316 号

责任编辑：董 英
印 刷：北京京科印刷有限公司
装 订：三河市皇庄路通装订厂
出版发行：电子工业出版社
北京市海淀区万寿路 173 信箱 邮编：100036
开 本：787×1092 1/16 印张：32 字数：678 千字
版 次：2011 年 5 月第 1 版
2014 年 1 月第 2 版
2016 年 4 月第 3 版
印 次：2016 年 4 月第 1 次印刷
印 数：3000 册 定价：79.00 元（含 DVD 光盘 1 张）

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888。

质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线：（010）88258888。

前　言

很多搜索相关的技术已经得到了初步的解决。在国内产业界也已经有很多公司掌握了基本的搜索开发技术并拥有专业的搜索技术开发人员。但是越来越多有价值的资讯对现有技术的处理能力仍然是一个挑战。

为了方便实践，需要有良好实现的代码作为参考。为了节约篇幅，书中的代码只是核心片段。本书相关代码的完整版本在附带光盘中可以找到。

作者罗刚在参加编写本书之前，还独立撰写过《自己动手写搜索引擎》一书，与王振东共同编写过《自己动手写网络爬虫》一书。经过 10 多年的技术积累以及猎兔搜索技术团队每年若干的研发投入，相信猎兔已经能够比以前做得更好。但越是深入接触客户的需求，越感觉到技术本身仍需要更多进步，才能满足实用的需要。写这本书也是考虑到，也许还需要更多的前进，才能使技术产生质的飞跃。

本书分为相关技术总体介绍部分、爬虫部分、全文检索部分、自然语言处理部分以及相关案例分析部分。

爬虫部分从基本的爬虫原理开始讲解，通过介绍优先级队列、宽度优先搜索等内容引领读者入门；然后根据当前风起云涌的云计算热潮，重点讲述了云计算的基本原理及其在搜索中的应用，以及 Web 图分析、信息抽取等内容；为了能够让读者更深入地了解爬虫，本书还介绍了有关爬虫的数据挖掘的内容。

全文检索部分重点介绍了搜索的基本原理与使用。主要介绍了开源软件实现 Lucene 以及 Solr。不仅介绍了如何使用这些开源软件，而且还介绍了其中的一些实现原理。Lucene 更高版本的改进指出了当前需要解决的问题，欢迎读者在了解基本原理后进行更深入的研究。

自然语言处理部分向来是笔者关注的重点，因为系统的智能化依赖于此。开发中文搜索离不开中文分词。开发任何自然语言的搜索也离不开对相应语言的处理。对自然语言的处理其实也可以用到对 Java 或 C 语言这样的机器语言的处理方法，只不过处理自然语言更难一点。

虽然本书的每个章节都已经用代码强化了实现细节，但是对于初学者来说，也许需要更多的案例来理解相关技术在真实场景中的用法。案例分析部分介绍了在地理信息系统领域和户外活动搜索领域的应用。股票应用案例待整理完整后再加入。

本书适合需要具体实现搜索引擎的程序员使用，对于信息检索等相关研究人员也有一定的参考价值，同时猎兔搜索技术团队也已经开发出以本书为基础的专门培训课程和商业软件。

高级开发人员也可以参加猎兔的培训或者创业团队。职场人员经常面临各种压力。选择猎兔培训，不是几个月学完以后就不再见面，而是给大家提供持久的支持。当以后需要再次找工作的时候，或者需要创业时，依然可以在这里找到支持。很多商业运营的大项目失败的代价太高，所以他们往往只招有多年开发经验的工程师。但是为了成长就不要怕犯错误，在培训时可以等学员犯了错误之后再告知正确答案。有经验的工程师也可以在这里学习到完整的技术体系。

感谢开源软件开发人员和家人、关心猎兔的老师和朋友、创业伙伴以及信赖猎兔软件的客户多年来的支持。读者可以通过 QQ（270954928）联系作者，或者加 QQ 群（166015123）讨论相关技术问题。参与本书编写的有罗刚、张子宪、张继红、罗庭亮、高丹丹、任通通、孙宽、何淑琴、徐友峰、张进威、刘宇、石田盈，在此一并表示感谢。让我们通过合作共赢为技术发展创造更好的生态环境。

编著者

目　录

第 1 章

搜索引擎总体结构

本章首先概要地介绍搜索引擎的总体结构和基本模块，然后介绍其中最核心的模块全文检索的基本原理。为了尽快普及搜索引擎开发技术，本章介绍的搜索引擎结构可以采用开源软件实现。为了通过实践来深入了解相关技术，本章中会介绍相关的开发环境。本书介绍的搜索技术使用 Java 编程语言实现，之所以没有采用性能可能会更好的 C/C++，是希望读者不仅能够快速完成相关的开发任务，而且可以把相关实践作为一个容易上手的游戏。另外，为了集中关注程序的基本逻辑，书中的 Java 代码去掉了一些错误和异常处理，实际可以运行的代码可以在本书附带的光盘中找到。在后面的各章中会深入探索搜索引擎的每个组成模块。

1.1 搜索引擎基本模块

一个最简单的搜索引擎由索引和搜索界面两部分组成，相对完整的搜索结构如图 1-1 所示。

实现按关键字快速搜索的方法是建立全文索引库，所以最基础的程序是管理全文索引库的程序。搜索的数据来源可以是互联网或者数据库，也可以是本地路径等。搜索引擎的基本模块结构如图 1-2 所示。

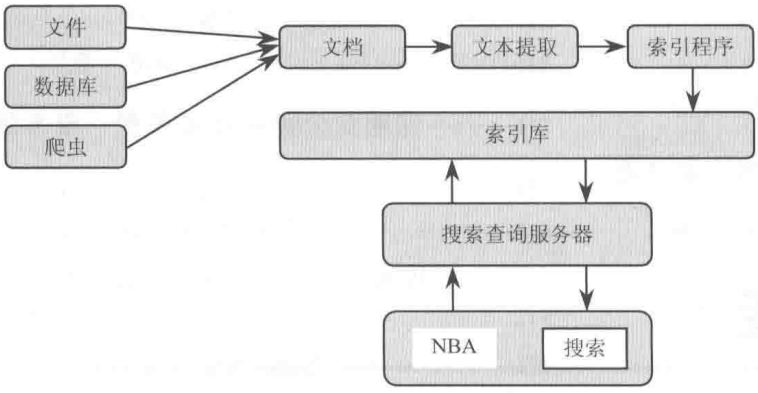


图 1-1　搜索引擎的简单结构

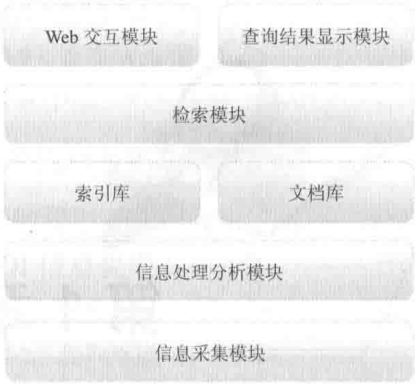


图 1-2　搜索引擎中的主要模块

1.2　开发环境

由于开源软件的迅速发展，可以借助开源软件简化搜索引擎开发工作。很多开源软件用 Java 语言开发，例如最流行的全文索引库 Lucene，所以本书采用 Java 来实现搜索。为了实现一个简单的指定目录文件的搜索引擎，首先要准备好 JDK 和集成开发环境 Eclipse。当前可以使用 JDK 1.8。JDK 1.8 可以从 Java 官方网站 http://www.oracle.com/technetwork/java/index.html 下载得到。使用默认方式安装即可。本书中的程序在附带光盘中都能找到，可以直接导入到 Eclipse 中。Eclipse 默认是英文界面，如果习惯用中文界面可以从 http://www.eclipse.org/babel/downloads.php 下载支持中文的语言包。

Lucene 是一个 Java 实现的 jar 包用来管理搜索引擎索引库。可以从 http://lucene.apache.org/core/下载到最新版本的 Lucene，当前的版本是 5.3.1。

如果需要用 Web 界面搜索，还要下载 Tomcat，当前可以从 http://tomcat.apache.org/下载到，推荐使用 Tomcat 6 以上的版本。使用开源的全文检索包 Lucene 做索引后，要把实现搜索的界面发布到 Tomcat。

Lucene 及一些相关项目的源代码由版本管理工具 SVN 管理，如果要构建源代码工程，可以使用工具 Ant 和 Maven。

如果需要导出 Lucene 的最新开发版本，就需要用到 SVN 的客户端。小乌龟 TortoiseSVN 是最流行的 SVN 客户端。TortoiseSVN 的下载地址是 http://tortoisesvn.tigris.org/。安装 TortoiseSVN 后，选择一个存放源代码的文件夹，单击鼠标右键，选择弹出菜单中的 Export

选项，导出源代码界面如图 1-3 所示。

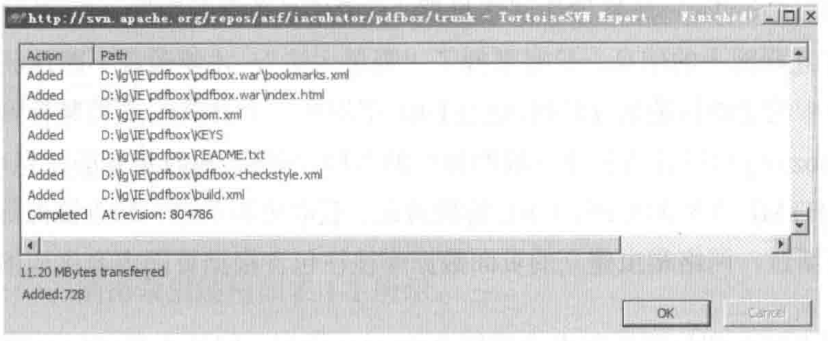


图 1-3　TortoiseSVN 导出代码界面

Ant 与 Maven 都和项目管理软件 make 类似。虽然 Maven 正在逐步替代 Ant，但当前仍然有很多开源项目在继续使用 Ant。从 http://ant.apache.org/bindownload.cgi 可以下载到 Ant 的最新版本。

在 Windows 下 ant.bat 和三个环境变量相关，即 ANT_HOME、CLASSPATH 和 JAVA_HOME。需要用路径设置 ANT_HOME 和 JAVA_HOME 环境变量，并且路径不要以\或/结束，不要设置 CLASSPATH。如果把 Ant 解压到 C:\apache-ant-1.7.1，则修改环境变量 PATH，增加当前路径 C:\apache-ant-1.7.1\bin。大部分用 Ant 构建的项目只需要如下命令即可：

```
#ant
```

可以从 http://maven.apache.org/download.html 下载最新版本的 Maven，当前版本是 maven-2.2.1。解压下载的 Maven 压缩文件到 C 盘根路径，将创建一个 C:\apache-maven-2.2.1 路径。修改 Windows 系统环境变量 PATH，增加当前路径 C:\apache-maven-2.2.1\bin。大部分用 Maven 构建的项目只需要如下命令即可：

```
#mvn clean install
```

1.3　搜索引擎工作原理

一个基本的搜索包括采集数据的爬虫和索引库管理以及搜索页面展现等部分。

1.3.1 网络爬虫

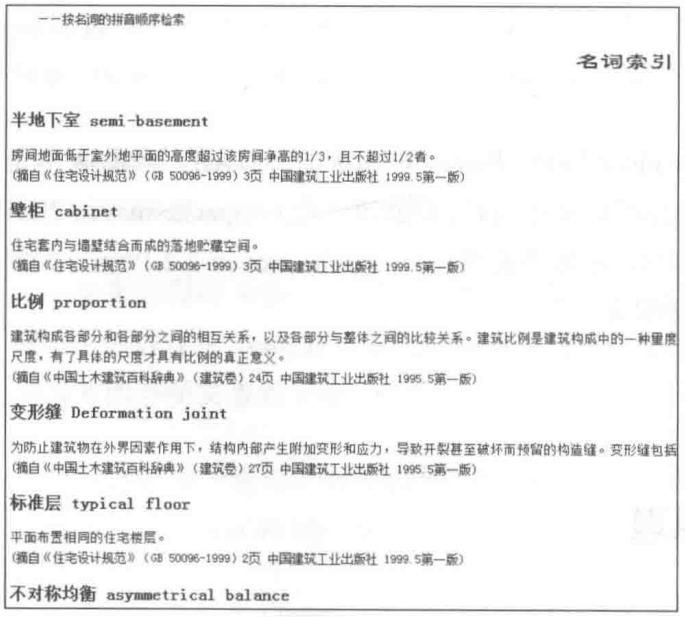
网络爬虫（Crawler）又被称作网络机器人（Robot）或者蜘蛛（Spider），它的主要目的是为获取在互联网上的信息。只有掌握了“吸星大法”，才能源源不断地获取信息。网络爬虫利用网页中的超链接遍历互联网，通过URL引用从一个HTML文档爬行到另一个HTML文档。http://dmoz.org 可以作为整个互联网抓取的入口。网络爬虫收集到的信息可有多种用途，如建立索引、HTML 文件的验证、URL 链接验证、获取更新信息、站点镜像等。为了检查网页内容是否更新过，网络爬虫建立的页面数据库往往包含根据页面内容生成的文摘。

在抓取网页时大部分网络爬虫会遵循 Robot.txt 协议。网站本身可以有两种方式声明不想被搜索引擎收入的内容：第一种方式是在站点的根目录下增加一个纯文本文件 http://www.yourdomain.com/robots.txt；另外一种方式是直接在 HTML 页面中使用 robots 的 meta 标签。

1.3.2 全文索引结构与 Lucene 实现

为了方便查询，早在计算机出现之前就出现了人工为图书建立的索引，比如图 1-4 中的名词索引。

为了按词快速定位抓取过来的文档，需要以词为基础建立全文索引，也叫倒排索引（Inverted index），如图 1-5 所示。


——按名词的拼音顺序检索

名词索引

半地下室 semi-basement

房间地面低于室外地平面的高度超过该房间净高的1/3，且不超过1/2者。
(摘自《住宅设计规范》（GB 50096-1999）3页 中国建筑工业出版社 1999.5第一版)

壁柜 cabinet

住宅套内与墙壁结合而成的落地贮藏空间。
(摘自《住宅设计规范》（GB 50096-1999）3页 中国建筑工业出版社 1999.5第一版)

比例 proportion

建筑构成各部分和各部分之间的相互关系，以及各部分与整体之间的比较关系。建筑比例是建筑构成中的一种重度尺度，有了具体的尺度才具有比例的真正意义。
(摘自《中国土木建筑百科辞典》（建筑卷）24页 中国建筑工业出版社 1995.5第一版)

变形缝 Deformation joint

为防止建筑物在外界因素作用下，结构内部产生附加变形和应力，导致开裂甚至破坏而预留的构造缝。变形缝包括
(摘自《中国土木建筑百科辞典》（建筑卷）27页 中国建筑工业出版社 1995.5第一版)

标准层 typical floor

平面布置相同的住宅楼层。
(摘自《住宅设计规范》（GB 50096-1999）2页 中国建筑工业出版社 1999.5第一版)

不对称均衡 asymmetrical balance

图 1-4 人工建立的名词索引

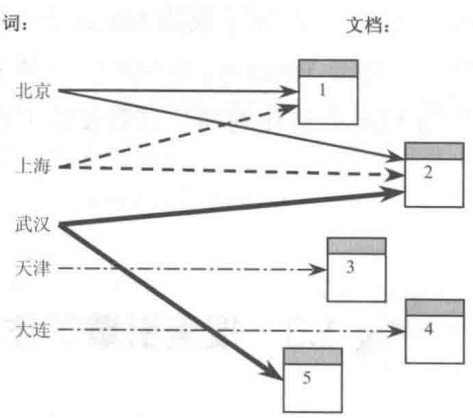


图 1-5 以词为基础的全文索引

倒排索引是相对于正向索引来说的，首先用正向索引来存储每个文档对应的单词列表，然后再建立倒排索引，根据单词来索引文档编号。

例如要索引如下两个文档：

```
Doc1：自己动手写搜索引擎
Doc2：自己动手写网络爬虫
```

在 Lucene 中的倒排索引结构如表 1-1 所示。

表 1-1　Lucene 中的倒排索引结构

词	(文档, 频率)	位　置
动手	(1, 1),(2, 1)	(2),(2)
搜索引擎	(1, 1)	(4)
网络爬虫	(2, 1)	(4)
写	(1, 1),(2, 1)	(3),(3)
自己	(1, 1),(2, 1)	(1),(1)

每个单词（term）后面的文档编号（docId）列表叫做投递列表（posting list）。在 Lucene 中，倒排索引结构存储在二进制格式的多个索引文件中，其中以 tis 为后缀的文件中包含了单词信息；以 frq 为后缀的文件记录单词的文档编号和这个单词在文档中出现了多少次，也就是频率信息；以 prx 为后缀的文件包含了单词出现的位置信息。

为了快速查找单词，可以先对单词列表排序，然后再折半查找已经排好序的词表。下面是实现折半查找的代码。

```
int low = fromIndex; //开始位置
int high = toIndex - 1; //结束位置

while (low <= high) {
    int mid = (low + high) >>> 1; //相当于mid = (low + high)/2
    Comparable midVal = (Comparable)a[mid]; //取中间的值
    int cmp = midVal.compareTo(key); //中间值和要找的关键字比较

    if (cmp < 0)
        low = mid + 1;
    else if (cmp > 0)
        high = mid - 1;
    else
        return mid; //查找成功，返回找到的位置
}
return -(low + 1);  //没找到，返回负值
```

词表是类似这样的数据结构：SortedMap<Term,Postings>。如果词表很小，内存能够放下，则可以使用折半查找算法来查询一个词对应的文档序列。如果内存不能完全放下倒排

索引中的词表，如何利用索引文件查找呢？理论上，可以采用 B 树存储词表。为了简化实现，Lucene3 以前的版本使用了跳表。跳表的实现把词组织成固定大小的块，例如每 32 个词放入一个新的块，然后在块上建立一个索引，记住每个块的开始词在文件中的位置。

Lucene（http://lucene.apache.org/）是一个开放源代码的全文索引库。经过 10 多年的发展，Lucene 拥有了大量的用户和活跃的开发团队。Eclipse 软件和 Twitter 网站等都在使用 Lucene。如果说 Google 是拥有最多用户访问的搜索引擎网站，那么拥有最多开发人员支持的搜索软件项目也许是 Lucene。

Lucene 的整体结构如图 1-6 所示。

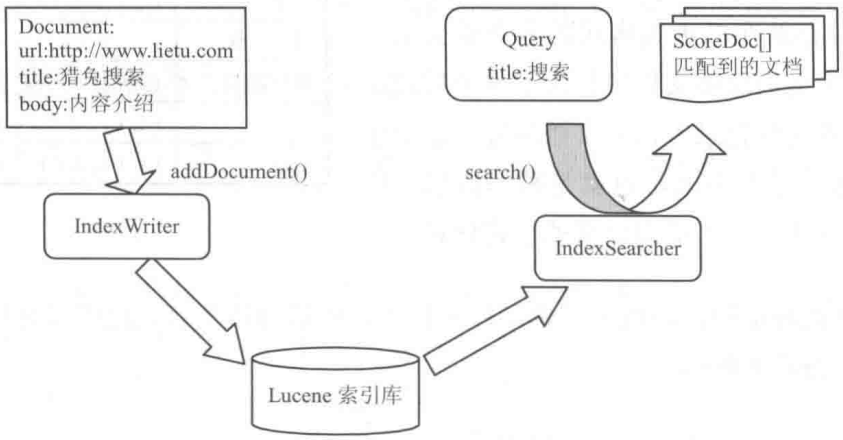


图 1-6　Lucene 原理图

Lucene 中的基本概念介绍如下。

- 第一个概念是 Index，也就是索引库。文档的集合组成索引。和一般的数据库不一样，Lucene 不支持定义主键。在 Lucene 中并不存在一个叫做 Index 的类。通过 IndexWriter 来写索引，通过 IndexReader 来读索引。索引库在物理形式上一般是位于一个路径下的一系列文件。
- 一段有意义的文字需要通过 Analyzer 分割成一个个词语后才能按关键词搜索。Analyzer 就是分析器，StandardAnalyzer 是 Lucene 中最常用的分析器。为了达到更好的搜索效果，不同的语言可以使用不同的分析器，例如 CnAnalyzer 就是一个主要处理中文的分析器。
- Analyzer 返回的结果就是一串 Token。Token 包含一个代表词本身含义的字符串和该词在文章中相应的起止偏移位置，Token 还包含一个用来存储词类型的字符串。
- 一个 Document 代表索引库中的一条记录，也叫做文档。要搜索的信息封装成

Document 后通过 IndexWriter 写入索引库。调用 Searcher 接口按关键词搜索后，返回的也是一个封装后的 Document 列表。

- 一个 Document 可以包含多个列，叫做 Field。例如一篇文章可以包含“标题”、“正文”、“修改时间”等 Field。创建这些列对象以后，可以通过 Document 的 add 方法增加这些列。和一般的数据库不一样，一个文档的一个列可以有多个值。例如一篇文档既可以属于互联网类，又可以属于科技类。
- Term 是搜索语法的最小单位，复杂的搜索语法会分解成一个 Term 查询。它表示文档的一个词语，Term 由两部分组成：它表示的词语和这个词语所出现的 Field。

Lucene 中的 API 相对数据库来说比较灵活，没有类似数据库先定义表结构后使用的过程。如果前后两次写索引时定义的列名称不一样，Lucene 会自动创建新的列，所以 Field 的一致性需要我们自己掌握。

1.3.3　搜索用户界面

随着搜索引擎技术逐渐走向成熟，搜索用户界面也形成了一些比较固定的模式。

- 输入提示词：用户在搜索框中输入查询词的过程中随时给予查询提示词。对中文来说，当用户输入拼音时，也能提示。
- 相关搜索提示词：当用户对当前搜索结果不满意时，也许换一个搜索词就能够得到更有用的信息。一般会根据用户当前搜索词给出多个相关的提示词。可以看成是协同过滤在搜索词上的一种具体应用。
- 相关文档：返回和搜索结果中的某一个文档相似的文档。例如：Google 搜索结果中的“类似结果”。
- 在结果中查询：如果返回结果很多，则用户在返回结果中再次输入查询词以缩小查询范围。
- 分类统计：返回搜索结果在类别中的分布图。用户可以按类别缩小搜索范围，或者在搜索结果中导航。有点类似数据仓库中的向下钻取和向上钻取。
- 搜索热词统计界面：往往按用户类别统计搜索词，例如按用户所属区域或者按用户所属部门等，当然也可以直接按用户统计搜索热词，例如 Google 的 Trends。

搜索界面的改进都是以用户体验为导向，所以搜索用户界面往往还会根据具体应用场景优化。所有这一切都是为了和用户的交互达到最优的效果。

1.3.4 计算框架

互联网搜索经常面临海量数据，需要分布式的计算框架来执行对网页重要度打分等计算。有的计算数据很少，但是计算量很大；还有些计算数据量比较大，但是计算量相对比较小。例如计算圆周率是计算密集型，互联网搜索中的计算往往是数据密集型，所以出现了数据密集型的云计算框架。MapReduce 是一种常用的云计算框架。

MapReduce 把计算任务分成两个阶段。映射（Map）阶段按数据分类完成基本计算；化简（Reduce）阶段收集基本的计算结果。使用 MapReduce 统计词频的例子如图 1-7 所示。

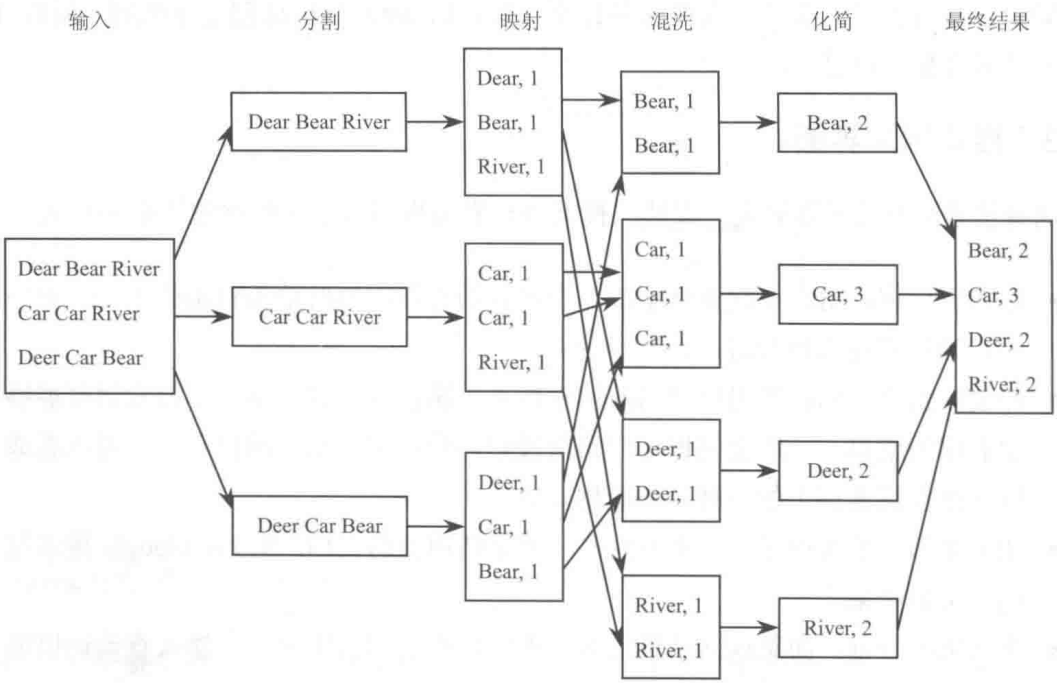


图 1-7 词频统计的例子

Hadoop（http://hadoop.apache.org/）是 MapReduce 思想实现的一个开源计算平台，已经在包括百度等搜索引擎开发公司得到商用。但是 MapReduce 是批处理的操作方式，一般来说，直到完成上一阶段的操作后才能启动下一阶段的操作。

我们需要有一种计算，可以尽快出结果，随着时间的延长，计算结果会越来越好。很多计算可以用迭代的方式做，迭代次数越多，结果往往越好，比如 PageRank 或者 KMeans、EM 算法。当然，这个应该不只需要迭代，还需要向最优解收敛。