- 从信息检索的核心概念入手，介绍Lucene与分布式搜索服务器Elasticsearch的相关知识
- 从原理到实践，涵盖Elasticsearch 5.4 开发所需要的各种技术
- 以实例为导向，并提供可执行程序与详尽的代码注解，有效降低学习门槛

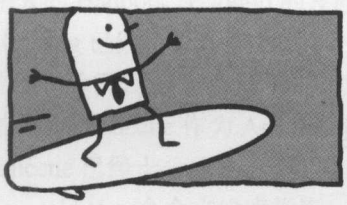
从Lucene到Elasticsearch

全文检索实战

姚 攀 编著

大数据时代的信息检索技术

清華大學出版社



从Lucene到Elasticsearch全文检索实战

姚 攀 编著

清华大学出版社
北京

内 容 简 介

本书循序渐进介绍了信息检索、布尔检索、向量空间模型、tf-idf、BM25 排序算法、Lucene 架构、Lucene 创建索引、Lucene 查询、Lucene 项目实战、Elasticsearch 安装与配置、Elasticsearch 插件安装、REST API 数据操作、映射与模板、索引别名、Elasticsearch 基本和高级搜索、Elasticsearch 同步数据库、Elasticsearch 集群管理、项目实战等内容。阅读本书，读者能够掌握信息检索的核心概念，应用 Lucene 库处理全文检索业务，掌握 Elasticsearch 分布式搜索引擎的使用方法与技巧。

本书基于 Lucene 6.0 和 Elasticsearch 5.4.0 进行讲解，技术先进，示例丰富，适合想学习信息检索技术的初学者和相关专业的大学生、研究生学习，也很适合大数据及云计算平台构建人员以及有一定基础的 IT 开发人员使用。

图书在版编目（CIP）数据

从 Lucene 到 Elasticsearch：全文检索实战/姚攀. —北京：清华大学出版社，2017
ISBN 978-7-302-48306-9

Ⅰ. ①从… Ⅱ. ①姚… Ⅲ. ①全文检索 Ⅳ. ①G254.923

中国版本图书馆 CIP 数据核字（2017）第 215137 号

责任编辑：王金柱
封面设计：王　翔
责任校对：闫秀华
责任印制：李红英

出版发行：清华大学出版社
网　　址：http://www.tup.com.cn，http://www.wqbook.com
地　　址：北京清华大学学研大厦 A 座　　邮　　编：100084
社 总 机：010-62770175　　邮　　购：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质 量 反 馈：010-62772015，zhiliang@tup.tsinghua.edu.cn

印 装 者：清华大学印刷厂
经　　销：全国新华书店
开　　本：190mm×260mm　　**印　　张**：20.5　　**字　　数**：525 千字
版　　次：2017 年 12 月第 1 版　　**印　　次**：2017 年 12 月第 1 次印刷
印　　数：1~3000
定　　价：79.00 元

产品编号：072144-01

前　言

我们正处在一个大数据时代，大数据并不仅仅是指海量数据，而更多的是指这些数据都是非结构化的、无法用传统的方法进行处理的数据。相信很多人听说过目前在云计算和大数据领域里如日中天的 Hadoop，Hadoop 的发起人之一是大名鼎鼎的 Doug Cutting。早在 Hadoop 诞生之前，Doug Cutting 已经用 Java 实现了第一个提供全文文本搜索的开源函数库 Lucene。Lucene 自 2000 年发布第一个开源版本以来，在开源社区引起了很大的反响，为广大开发者提供了研发全文检索系统的利器。Lucene 作为 Apache 的顶级项目，有大量研发人员贡献源码，经过十几年的发展，目前 Lucene 已经十分成熟，可以说 Lucene 是当今最先进、最高效的全功能开源搜索引擎工具包。但 Lucene 只是一个全文检索类库，Elasticsearch 是一个建立在 Lucene 基础上的实时的分布式搜索引擎，2010 年由 Shay Bano 发布。相比于 Lucene，Elasticsearch 功能更加强大，使用更加方便。

站在巨人的肩膀上，入门搜索技术并不困难，本书为入门 Lucene、Elasticsearch 而生。本书首先介绍信息检索领域中的一些基本理论，也就是 Lucene 的数学模型，之后介绍如何使用 Lucene 库构建全文检索系统，最后介绍 Elasticsearch。本书按照从数学模型到入门基础再到项目实战的思路来编写，数学模型让读者知其然也知其所以然，入门基础是理论到实际应用的必经之路，项目实战则是为了学以致用。书中的每一部分都力图简明扼要，使用大量实例和代码，为读者能够快速掌握全文检索技术扫除障碍。将全文检索领域中的一些知识和项目经验分享给大家，是笔者写作本书的初衷。

本书结构

本书从逻辑上可划分为三部分。

第一部分（第 1 章），主要介绍信息过载、信息检索、倒排索引、布尔模型、tf-idf、向量空间模型、概率检索模型等信息检索领域的基础知识。

第二部分（第 2 和 3 章），介绍如何使用 Lucene 开发全文检索系统。

第 2 章主要介绍 Lucene 的基础知识，内容包括 Lucene 的特点、Lucene 架构、Luke 的使用、IK 分词器配置、扩展词库和远程词库的配置、Lucene 的多种分词器、索引的构建方法、检索文档以及实现检索关键词高亮的方法。

第 3 章是 Lucene 项目实战部分，介绍如何使用 Lucene 构建一个文件检索系统，内容包括项目的整体设计、使用 Tika 做信息抽取、索引的构建、用户查询界面的设计与实现、用户查询处理、搜索结果展示等内容。

第三部分（第 4~11 章），主要介绍 Elasticsearch 分布式搜索引擎的相关技术。

第 4 章是 Elasticsearch 简介，内容包括 Elasticsearch 与 Lucene 的关系、Elasticsearch 的整体架构、核心概念、在企业中的应用案例、流行度趋势、Elasticsearch 的安装、中文分词配置以及相关插件的安装与使用。

第 5 章是 Elasticsearch 集群入门，主要内容包括索引管理、文档管理和映射详解。

第 6 章介绍 Elasticsearch 的搜索功能，主要内容包括搜索机制的解读、全文查询、词项查询、复合查询、嵌套查询、位置查询、特殊查询、搜索高亮和排序。

第 7 章介绍 Elasticsearch 的聚合分析功能。

第 8 章介绍如何使用 Elasticsearch Java API 做二次开发。

第 9 章介绍 Elasticsearch 集群管理的相关知识点，包括脑裂问题、集群规划、索引规划、分布式集群的搭建方法以及如何查看集群的监控信息。

第 10 章是 Elasticsearch 整合 MySQL 项目实战部分，通过实现对 MySQL 中的数据进行全文检索这一需求，贯穿了 MySQL、JDBC、Elasticsearch Java API 以及 Java Web 的相关知识，使读者了解在实际的项目开发中使用 Elasticsearch 做全文搜索的方法。

第 11 章介绍 Elasticsearch 和 Hadoop 大数据平台交互的方法。

学习本书的预备知识

Java 基础

首先要配置好 Java 开发环境。不论是学习 Lucene 还是 Elasticsearch 都需要安装好 Java 环境，Elasticsearch 的运行要求 JDK 版本最低为 1.7，建议使用 JDK 1.8 及以上版本。鉴于 Java 的跨平台特性，对操作系统没有要求，在 Windows、Linux、Mac OS X 系统上都可以运行 Elasticsearch。除此之外，读者需要掌握 Java 基础知识。

Java Web 开发技术

在项目实战中需要用到 Java Web 的相关技术，建议读者在阅读本书之前掌握 HTML、CSS、JSP 等基础知识，掌握 Java Web 项目的部署和运行。

本书使用的软件版本

本书基于 Lucene 6.0 和 Elasticsearch 5.4.0 进行讲解，集成开发环境为 Eclipse 4.6.1。

读者对象

在校学生

如果你是正在大学校园里修读计算机科学相关专业的大学生，也许你正在选修程序设计语言，课程结束了你发现自己只能写出命令行下黑白屏显示的小程序，你也许很期待学到更多的技术做出实际的项目，那么本书就是为你准备的。书中的项目使用的是 Java 语言，除了 Lucene 和 Elasticsearch 的使用之外，还穿插了 Java SE、Java Web 的相关技术。

Java 程序开发者

如果你是已经参加工作的 Java 程序开发者，想要掌握全文检索相关技术却不知道从哪里入手，需要处理企业中的全文检索业务却没有思路，你也许听说过 Lucene 或 Elasticsearch，但是不知道怎样快速入门，那么本书可以作为入门全文检索、学习 Lucene 和 Elasticsearch 开发技术的参考书。

搜索引擎研发人员

如果你是搜索引擎研发者，本书中的实际案例和相关知识点可以作为参考资料，比如信息检索模型理论基础、文档信息抽取、Lucene 应用案例、Elasticsearch Java API、Elasticsearch 集群管理等。希望能以本书为媒介和大家共同探讨和交流。

源代码下载

源代码下载地址：http://pan.baidu.com/s/1slHRM5f（注意区分数字和英文字母的大小写）

勘误与交流

限于笔者水平和写作时间有限，不可避免地会有些疏漏之处，欢迎大家通过电子邮件等方式批评指正。

笔者的邮箱：ucasyp@163.com

笔者的博客：blog.csdn.net/napoay

致谢

本书能够顺利出版要感谢很多单位和个人。首先要感谢笔者的家人，他们对笔者学业的支持和生活的照顾使笔者没有后顾之忧，全身心投入到本书的写作当中。

感谢北京博瑞开源有限公司，公司给笔者提供了宝贵的实习机会，本书的很多知识点都来源于实际项目，是在解决实际问题过程中的经验总结，感谢公司董事长李小翔先生、架构师黄超对笔者的指导和帮助。

感谢马玉鹏老师、郎睿师兄、张港红博士、CSDN 博主周程（blog.csdn.net/fxsdbt520）、秦雪箭、宗鹏、陆风光在本书写作过程中的帮助和支持。

感谢清华大学出版社给了笔者一次和大家分享技术、交流学习的机会，感谢王金柱编辑在本书出版过程的辛勤付出。

姚　攀

2017 年 10 月 9 日

目　录

第 1 章

信息检索模型

本章学习要点：

* 信息过载简介
* 信息检索定义
* 分词算法简介
* 倒排索引介绍
* 布尔检索模型
* tf-idf 权重计算
* 向量空间模型
* 概率检索模型

1.1 信息检索概述

1.1.1 信息过载

互联网的飞速发展使人类进入了信息大爆炸的时代，根据相关统计数据显示，目前全球网民数量已经达到 32 亿人，互联网上的数据也呈指数级增长。图 1-1 是国外初创公司 Demo 发布的一张信息图，该图展示了各大网站在 60 秒内产生的巨大数据量。

根据 Demo 的数据显示，在一分钟内 YouTube 用户每分钟会上传 400 个小时的新视频，Netflix 用户每分钟则观看 86 805 个小时的视频。与此同时，苹果用户每分钟下载 51 000 个应用，亚马逊每分钟交易额达 222 283 美元，Google 在一分钟内翻译了 69 500 000 个单词，SIRI 一分钟内回答了 99 206 个问题，The Weather Channel（注：一款天气预报软件）每分钟接收 13 888 889 次天气查询请求。在社交网站方面，Facebook 用户每分钟分享 216 302 张照片，Dropbox 用户每分钟上传 833 333 个新文件，Tinder 用户每分钟发布 9678 条表情符号推文，而 Snapchat 用户每分钟会发布 284 722 张照片。

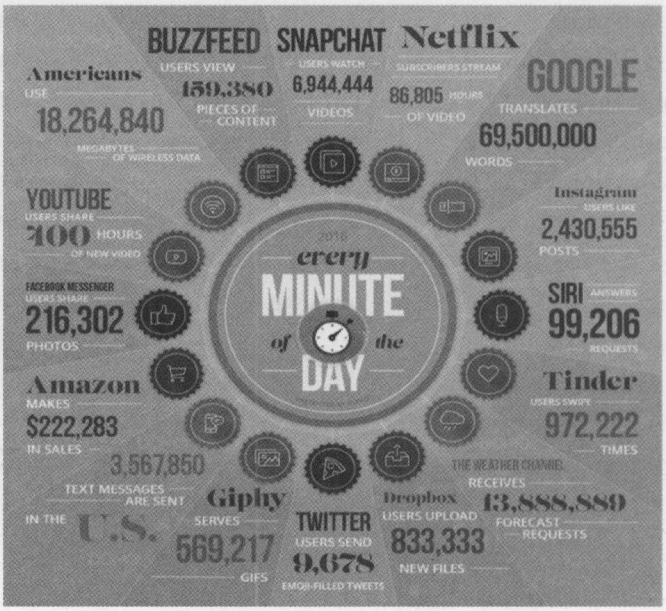


图 1-1　互联网上的一分钟

就在你看完上面这段内容的时间里，所有的一切都可能发生了改变。我们处在一个大数据时代，也是一个信息过载（Information Overload）的时代。大数据时代的特点可以用 4 个 V 来概括：

- Volume
  数据量大，全球每年产生的数据总量已经达到 ZB（1ZB=2^{40}GB）级别。
- Variety
  数据种类繁多，如文本、图片、视频、地理信息、各种传感器信息等。
- Velocity
  数据流动速度快，对数据处理的时效性要求高。
- Value
  大数据蕴含着巨大的价值，可以帮助人们解决数据量不足时所不能解决的问题。

信息过载是指社会信息超过了个人或系统所能接受、处理或有效利用的范围，并导致故障的状况。信息过载主要有以下 3 个特点：

（1）受传者对信息反映的速度远远低于信息传播的速度。

（2）大众媒介中的信息量大大高于受众所能消费、承受或需要的信息量。

（3）大量无关的、没用的、冗余的信息严重干扰了受众对相关有用信息的准确分析和正确选择。

信息过载是信息时代信息极大丰富的负面影响之一。

1.1.2　信息检索定义

信息资源总量呈爆炸式增长，在信息的海洋里获取想要的信息变得更加困难。为了解决

信息过载的问题，无数科学家和工程师提出了很多天才的解决方案，其中最具代表性的是分类目录和搜索引擎。

分类目录是将网站信息系统地分类整理，提供一个按类别编排的网站目录，在每类中排列着属于这一类别的网站站名、网址链接、内容提要以及子分类目录，可以在分类目录中逐级浏览寻找相关的网站，分类目录中往往还提供交叉索引，从而可以方便地在相关的目录之间跳转和浏览。互联网早期的门户网站，比如雅虎、搜狐、新浪等，都是将不同来源的信息以一种整齐划一的形式整理、储存并呈现给用户，用户根据信息来源、信息类型、关键字等方式筛选网站内容。

搜索引擎是指自动从因特网搜集信息，经过一定整理以后，提供给用户进行查询的系统。国外具有代表性的搜索引擎有 Google、Bing、Yahoo 等，国内具有代表性的搜索引擎有百度搜索、搜狗搜索、360 搜索等。

我们常用的搜索引擎是 Web 搜索，是信息检索的一个分支，学术上的信息检索（Information Retrieval，简称 IR）的定义为：信息检索是从大规模非结构化数据（通常是文本）的集合（通常保存在计算机上）中找出满足用户信息需求的资料（通常是文档）的过程。

1.1.3　信息检索常用术语

信息检索领域有一些常用的术语，深刻理解这些术语对入门信息检索非常有必要，简介如下。

- 用户需求（User Need，简称 UN）
  用户需要获得的信息。严格地说，UN 只存在于用户的内心，但是通常用文本来描述，如查找与“2014 世界杯”相关的新闻，有时也称为主题（Topic）。
- 查询（Query）
  UN 提交给检索系统时称为查询（Query），如“iPhone7 价格”。对同一个 UN，不同人不同时候可以构造出不同的 Query，上述需求也可表示成“苹果 7 价格”。Query 在 IR 系统中往往还有内部表示。
- 文档（Document）
  文档是信息检索的对象，文档不仅仅可以是文本，也可以是图像、视频、语音等多媒体文档。
- 文档集（Crops）
  由若干文档构成的集合称为文档集合，文档集有时也称为语料库。海量的互联网网页、文件系统中的文本文件、大量的电子邮件，都是文档集合的具体例子。
- 文档编号（Document ID）
  文档 ID 是给文档集中的每个文档赋予的唯一标识符，通过文档 ID 来区分不同的文档，这样能够方便搜索引擎的内部处理。缩写为 docID。
- 词条化（tokenization）
  词条化是将给定的字符序列拆分成一系列子序列的过程，拆分的每个子序列称为一个词条。词条化的过程中有可能会去除标点符号等特殊字符。下面是一个词条化的具体例子。

  输入：Whatever happens tomorrow,we have had today.
  输出：[whatever] [happens] [tomorrow] [we] [have] [had] [today]

- 词项（Term）

  词项是经过语言学预处理之后归一化的词条。词项是索引的最小单位，一般情况下可以把词项当作词，但词项不一定就是词。对于上面的句子，产生词项如下：

  whatever　happen　tomorrow　we　have　had　today

- 词项-文档关联矩阵（Incidence matrix）

  词项-文档关联矩阵是表示词项和文档之间所具有的一种包含关系的概念模型，表 1-1 展示了其含义。表中的每列代表一个文档，每行代表一个词项，打对勾的位置代表包含关系。

表1-1　词项－文档关联矩阵

	doc1	doc2	doc3	doc4	doc5	doc6
term1	√		√			√
term2		√			√	
term3		√		√		
term4	√		√		√	
term5	√			√		√
term6		√	√		√	

从纵向即文档这个维度来看，每列代表一个文档包含的词项信息，比如 doc1 包含了 term1、term4 和 term5，而不包含 term2、term3、term6。从横向即词项这个维度来看，每行代表该词项在文档中的分布信息，比如对于 term1 来说，doc1、doc3、doc6 中出现过 term1，而其他文档不包含 term1。矩阵中其他的行列也可作此种解读。

- 词项频率（Term frequency）

  同一个单词在某个文档中出现的频率。比如，单“apple”在某文档中出现了 3 次，那么该单词在该文档中的词项频率就是 3。

- 文档频率（Document frequency）

  出现某词项的文档的数目。比如，单词“China”只出现在文档集合中的文档 1 和文档 5，那么该单词的文档频率就是 2。

- 倒排记录表（Postings lists）

  倒排记录表用于记录出现过某个单词的所有文档的文档列表以及单词在该文档中出现的位置信息，每条记录称为一个倒排项。通过倒排列表即可获知哪些文档包含哪些单词。

- 倒排文件（Inverted file）

  倒排记录表在磁盘中的物理存储文件称为倒排文件。

1.1.4　信息检索系统

一个完整的信息检索系统的基本架构如图 1-2 所示。信息检索系统可以分为信息采集、信息整理和用户查询 3 部分。

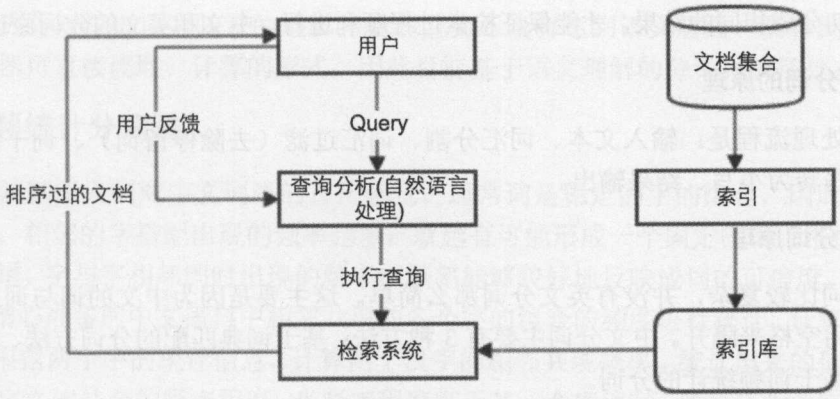


图 1-2　IR 系统基本架构图

1. 信息采集

信息采集基本都是通过网络爬虫（Spider）自动完成的。网络爬虫是一种按照一定的规则，自动地抓取万维网信息的程序或者脚本。互联网上的网页数以亿计，遍布在全球的各个服务器上，通过爬虫可以将网页下载下来进行进一步的分析和挖掘，经过格式处理之后提取网页信息为构建索引做准备。

2. 整理信息

信息检索系统整理信息的过程称为索引构建。信息检索系统不仅要保存搜集起来的信息，还要将它们按照一定的规则进行编排，这样就不用重新翻查它所有保存的信息就能迅速找到所要的资料。如果信息是不按任何规则地随意堆放在系统中，那么它每次找资料都得把整个资料库完全翻查一遍，如此一来再快的计算机系统也没有用。

3. 接受查询

用户向信息检索系统发出查询请求，信息检索系统接受查询并向用户返回检索到的文档。信息检索系统（尤其是商用搜索引擎）每时每刻都要接到来自大量用户的几乎是同时发出的查询，它按照每个用户的要求检查自己的索引，在极短时间内找到用户需要的文档，并返回给用户。目前，搜索引擎返回主要是以网页链接的形式提供的，通过这些链接用户便能到达含有自己所需资料的网页。搜索引擎通常会在这些链接下提供一小段来自这些网页的摘要信息以帮助用户判断此网页是否含有自己需要的内容。

1.2　分词算法

1.2.1　分词算法概述

词是表达语义的最小单位。分词对搜索引擎的帮助很大，可以帮助搜索引擎程序自动识别语句的含义，从而使搜索结果的匹配度达到最高，因此分词的质量也就直接影响了搜索结果的精确度。分词在文本索引的建立过程和用户提交检索过程中都存在。利用相同的分词器，把短

语或者句子切分成相同的结果，才能保证检索过程顺利进行。中文和英文的分词原理简介如下：

1. 英文分词的原理

基本的处理流程是：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）、大写转为小写、结果输出。

2. 中文分词原理

中文分词比较复杂，并没有英文分词那么简单。这主要是因为中文的词与词之间并不像英文中那样用空格来隔开。中文分词主要有 3 种方法：基于词典匹配的分词方法、基于语义理解的分词、基于词频统计的分词。

1.2.2　词典匹配分词法

基于字典匹配的分词方法按照一定的匹配策略将输入的字符串与机器字典词条进行匹配，这种方法是最简单的也是最容易想到的分词办法，最早由北京航空航天大学的梁南元教授提出。查字典分词实际上就是把一个句子从左向右扫描一遍，遇到字典中有的词就标识出来，遇到复合词就找到最长的词匹配，遇到不认识的字串则切分成单个词。按照匹配操作的扫描方向不同，字典匹配分词方法可以分为正向匹配、逆向匹配以及结合了两者的双向匹配算法；按照不同长度优先匹配的情况，可以分为最大（最长）匹配和最小（最短）匹配；按照是否与词性标注过程相结合，又可以分为单纯分词方法和分词与词性标注相结合的方法。几种常用的词典分词方法如下：

- 正向最大匹配（由左到右的方向）
- 逆向最大匹配（由右到左的方向）
- 最少切分（是每一句中切除的词数最小）

实际应用中上述各种方法经常组合使用，以达到最好的效果，从而衍生出了结合正向最大匹配方法和逆向最大匹配算法的双向匹配分词法。由于中文分词最大的问题是歧义处理，结合中文语言自身的特点，经常采用逆向匹配的切分算法，处理的精度高于正向匹配，产生的切分歧义现象也较少。

真正实用的分词系统，都是把词典分词作为基础手段，结合各种语言的其他特征信息来提高切分的效果和准确度。有的实用系统中将分词和词性标注结合起来，利用句法和词法分析对分词决策提高帮助，在词性标注过程中迭代处理，利用词性和语法信息对分词结果进行检验、调整。

1.2.3　语义理解分词法

基于语义理解的分词方法是模拟人脑对语言和句子的理解，达到识别词汇单元的效果。基本模式是把分词、句法、语义分析并行进行，利用句法和语义信息来处理分词的歧义。

一般结构中通常包括分词子系统、句法语义子系统、调度系统。在调度系统的协调下，分词子系统可以获得有关词、句子等的句法和语义信息，模拟人脑对句子的理解过程。基于语义理解的分词方法需要使用大量的语言知识和信息。

目前国内外对汉语语言知识的理解和处理能力还没有达到语义层面，具体到语言信息很难组织成机器可直接读取、计算的形式，因此目前基于语义理解的分词系统还处在试验阶段。

1.2.4　词频统计分词法

这种做法基于人们对中文词语的直接感觉。通常词是稳定的字的组合，因此在中文文章的上下文中，相邻的字搭配出现的频率越多，就越有可能形成一个固定的词。根据 n 元语法知识可以知道，字与字相邻同时出现的频率或概率能够较好地反映成词的可信度。实际的系统中，通过对精心准备的中文语料中相邻共现的各个字的组合的频度进行统计，计算不同字词的共现信息。根据两个字的统计信息，计算两个汉字的相邻共现概率，统计出来的信息体现了中文环境下汉字之间结合的紧密程度。当紧密程度高于某一个阈值时，便可认为此字组可能构成一个词。

基于词频统计的分词方法只需要对语料中的字组频度进行统计，不需要切分词典，因而又叫作无词典分词法或统计分词方法。这种方法经常抽出一些共现频度高但并不是词的常用字组，需要专门处理，提高精确度。实际应用的统计分词系统都使用一个基本的常用词词典，把字典分词和统计分词结合使用。基于统计的方法能很好地解决词典未收录新词的处理问题，即将中文分词中的串频统计和串匹配结合起来，既发挥匹配分词切分速度快、效率高的特点，又利用了无词典分词结合上下文识别生词、自动消除歧义的优点。

1.3　倒排索引

索引是构成搜索引擎的核心技术之一，索引在日常生活中其实也是非常常见的，比如当我们看一本书的时候，我们首先会看书的目录，通过目录可以快速定位到某一章节的页码，加快对内容的查询速度。

文档通常保存在各种数据库管理系统之中，比如 Oracle、MySQL 等。但是搜索引擎中的数据不能保存到数据库中，主要是因为数据库不能满足搜索引擎的需求，原因有二：一是搜索引擎中的数据量非常庞大，大型商业搜索引擎需要处理数以亿计的网页，面对海量数据使用关系型数据库很难管理；二是搜索引擎使用的数据操作非常简单，一般只需增删改查这几个基本功能，一般的数据库系统则支持大而全的功能，损失了速度和空间，大量用户检索则要求搜索引擎响应时间必须很快，检索效率要非常高，数据库系统在检索响应时间和检索并发度方面都不能满足需求。而数据库中的索引就是为了提高表的搜索效率而对某些字段中的值建立的目录，在搜索引擎中使用倒排索引这种数据结构来存储网页信息。

倒排索引（Inverted index），也常被称为反向索引，是一种索引方法，被用来存储在全文搜索下某个单词在一个文档或者一组文档中的存储位置的映射，它是文档检索系统中最常用的数据结构。

下面以简单通俗的例子来理解倒排索引，假设现在有两个文档 doc1 和 doc2，doc1 包含 3 个关键词：中国、美国、韩国，doc2 中包含 4 个关键词：中国、美国、德国、英国，文档和词语的包含关系（也就是正排索引），见表 1-2。