搜索引擎零距离

——基于Ruby+Java搜索引擎原理与实现

王 亮 编著

- 国内垂直搜索引擎的扛鼎之作
- 集开源搜索引擎之大成，融会贯通，自成一体
- 无线搜索引擎核心技术零距离接触
- Web信息挖掘专用程序设计语言，语法标准首次发布
- 垂直爬虫专用并行虚拟机核心技术展示
- 多年商业搜索引擎开发运营经验之提炼总结
- 真实的中型分布式搜索引擎开发案例全景展现
- 最新Java前沿技术在经典计算机理论上的优秀应用
- 专业信息检索理论与商业搜索需求的完美结合
- Java软件工程设计模式最佳实践

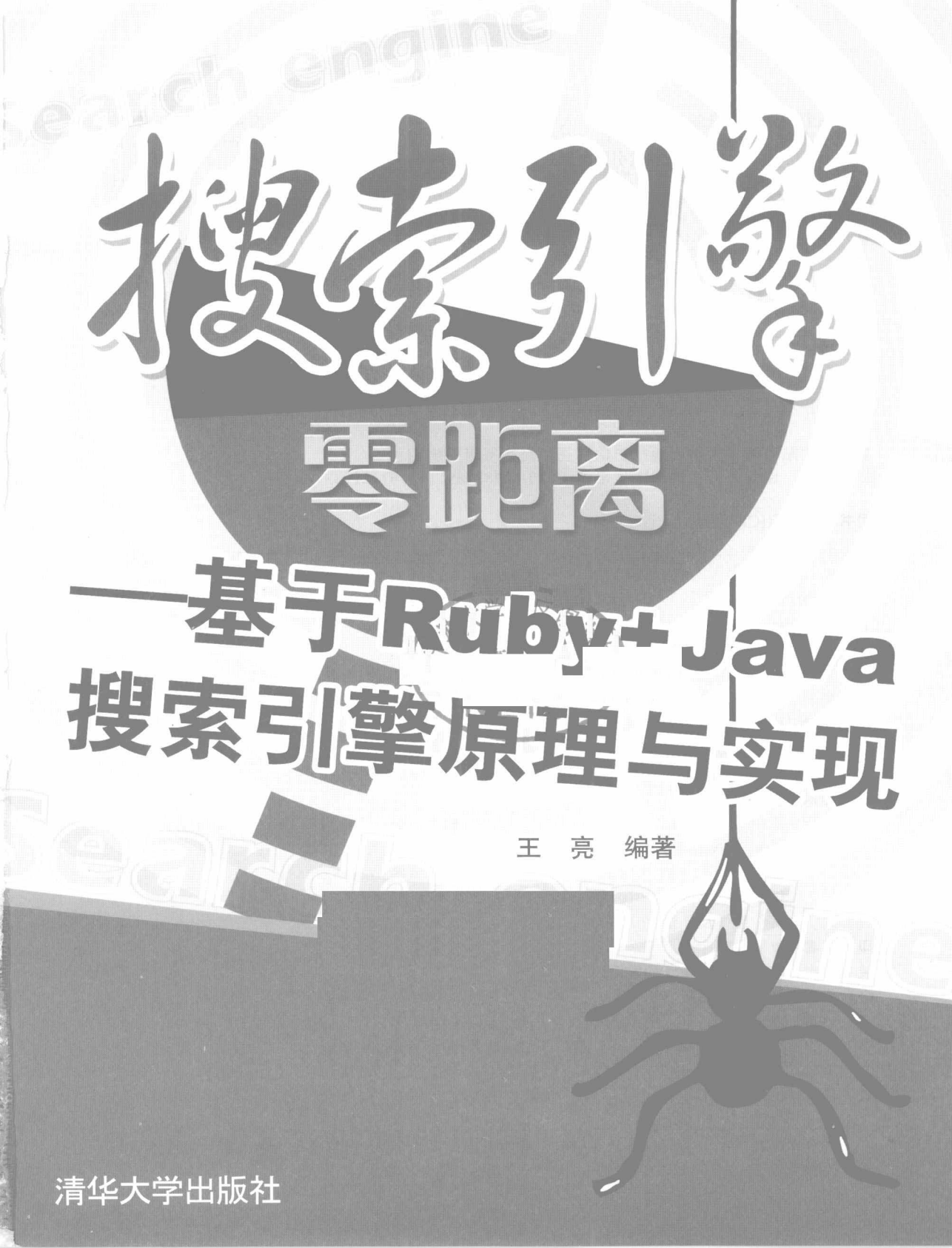
清华大学出版社

搜索引擎

零距离

——基于Ruby+Java搜索引擎原理与实现

王 亮 编著



清华大学出版社

内 容 简 介

随着网络信息资源的急剧增长，人们越来越多地关注如何快速有效地从海量的网络信息中，抽取出潜在的、有价值的信息，使之有效地在管理和决策中发挥作用。搜索引擎技术解决了用户检索网络信息的困难，目前搜索引擎技术正成为计算机科学界和信息产业界争相研究、开发的对象。

本书的作者是一位资深的搜索引擎开发人员，书中对数据获取(网络信息挖掘)与数据检索(搜索引擎)两个方面作了深入的介绍。本书首先提出了一套“网络数据挖掘”的完整理论，并给出一个实际的智能爬虫系统，通过理论与实际的完整呈现，使读者能够对“网络数据挖掘”有一个比较具体的认识，然后介绍了一个专用程序语言 IRS，并给出了这个语言的编译器以及虚拟机的实现方法。本书还通过对多个开源搜索引擎项目抽丝剥茧的细致分析，引出搜索引擎的一些基本原理与开发方法，并介绍了一个商业化搜索引擎的实例。本书的最后还结合一个 Java 框架介绍了一些软件设计思想。

本书涉及网络数据挖掘、搜索引擎原理、编译原理、数据库原理、正则表达式、软件工程、设计模式、Ruby 语言、HTTP 协议等计算机科学与技术的知识，适合搜索引擎开发人员作为参考，也适合有一定计算机基础的读者阅读，以扩展视野。

本书的内容中，既有教科书式的理论阐述，也有“七天入门”式的实例解析，还有《Linux 内核情景分析》风格的细致的代码分析，甚至还有一些英语文献翻译，从初学者到有一定经验的搜索引擎开发人员，各个层次的读者都能找到一些适合自己阅读的章节。

图书在版编目(CIP)数据

搜索引擎零距离——基于 Ruby+Java 搜索引擎原理与实现/王亮编著. —北京：清华大学出版社，2009.6
ISBN 978-7-302-20147-2

Ⅰ. 搜…　Ⅱ. 王…　Ⅲ. 互联网络—情报检索　Ⅳ. G354.4

中国版本图书馆 CIP 数据核字(2009)第 071698 号

责任编辑：邹　杰　桑任松
封面设计：山鹰工作室
版式设计：北京东方人华科技有限公司
责任校对：王　晖
责任印制：王秀菊
出版发行：清华大学出版社　　**地　　址**：北京清华大学学研大厦 A 座
http://www.tup.com.cn　　**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 刷 者：清华大学印刷厂
装 订 者：北京鑫海金澳胶印有限公司
经　　销：全国新华书店
开　　本：178×232　**印　张**：25.25　**字　数**：490 千字
版　　次：2009 年 6 月第 1 版　　**印　　次**：2009 年 6 月第1 次印刷
印　　数：1～4000
定　　价：39.80 元

前　言

搜索引擎是指因特网上专门提供查询服务的一类网站，这些网站通过网络搜索软件(又称为网络搜索机器人)或网站登录等方式，收集因特网上大量网站的页面，经过加工处理后建库，从而能够对用户提出的各种查询作出响应，提供用户所需的信息。用户的查询途径主要包括自由词、全文检索、主题词检索、分类检索及其他特殊信息的检索(企业、人名、电话黄页等)。

本书中所记述的种种理论与知识，是笔者在多年的搜索引擎开发过程中的积累与沉淀，其中既涉及了 Google 和 Baidu 这种类型的通用搜索引擎，也涉及了垂直搜索引擎的开发技术。

垂直搜索引擎概念的提出，就是针对某一特定领域、某一特定人群或某一特定需求提供的有一定价值的信息和相关服务。可以简单的说成是搜索引擎领域的行业化分工。众多专业性网站、行业网站独立服务于互联网的成功，恰恰证明了互联网的格局应该是多方面的。通用搜索引擎的性质，决定了其不能满足特殊领域和特殊人群的精准化信息需求服务。市场需求多元化决定了搜索引擎的服务模式必将出现细分，以针对不同行业提供更加精确的行业服务模式。可以说通用搜索引擎的发展为垂直搜索引擎的出现提供了良好的市场空间，势必将出现垂直搜索引擎在互联网中占据部分市场的趋势，也是搜索引擎行业细分化的必然趋势。

目前，我们已经可以在互联网上看到各种各样类型的垂直搜索网站，旅游、汽车、工作、房产，交友、车票、工作、股票、博客，这些网站提供的垂直搜索服务能够为用户提供比通用搜索引擎(Google 和 Baidu)更为准确、及时、全面的信息，使人们的生活更加方便。

多年来，笔者作为一名搜索引擎开发人员，参与和主持开发了多套搜索引擎系统，通过这些经历总结出了一些理论与经验，希望本书能对搜索引擎开发人员有所启发，帮助那些对搜索引擎开发有兴趣的读者踏进这扇大门，并为那些经验不那么丰富的搜索引擎开发者提供一些参考，使得广大互联网用户能在 Google 和 Baidu 之外的众多的垂直搜索网站上获得更好的体验与服务。

本书共分为 7 章，各章内容如下。

第 1 章　本章介绍网络数据挖掘的定义，并介绍了从网页中获得结构化数据的基本方法。网络数据挖掘是一个很大的话题，本书只涉及其中的一个领域，就是“网页数据挖掘”，也就是如何从互联网上“浩瀚无边”的万亿个网页中获取我们所需的结构化数据。

第 2 章　本章介绍一个“网页数据挖掘系统”(本书称之为“智能网络爬虫”)，

定义了这个“结构化信息采集与分析系统”的系统功能，包括入口定义、数据规格定义、页面发现、JavaScript 解析、模拟登录等基本功能。

第 3 章 本章介绍网页信息挖掘专用程序设计语言，在本书中称之为 IRS 语言(Information Retrival Script)，详细描述了这个语言的语法与功能特点，并给出了几个有详细运行流程描述的代码实例。

第 4 章 本章介绍 IRS 语言的虚拟机及编译器实现原理。先介绍了 IRS 编译器源代码进行词法、语法、语义分析的过程，然后描述了虚拟机解释执行 IRS 代码的过程。本章还简单介绍了内嵌的 JRuby 引擎以及 Ruby 语言的语法，并在解释了什么是“编译原理”之后比较详细地介绍了一个编译器生成框架 SableCC。

第 5 章 本章由浅入深地介绍搜索引擎设计原理，介绍了“倒排索引”、“反向文档频率”等搜索引擎专用术语，并介绍了多个著名的开源的搜索引擎相关的软件项目：Lucene、Hadoop、Nutch、Solr 和 Compass，及这些工具的使用方法，并对这些项目的一些主要代码进行了详细的情景分析。

第 6 章 本章介绍一个搜索引擎的商业化实现，也就是一个能够商业化运营的搜索引擎。本章通过对 Solr 和 MySE 这两个系统从灵活性、稳定性、性能扩展性、功能完善性等特点进行对比，介绍了一个完整的搜索引擎系统需要实现的各个功能点以及实现方法。

第 7 章 本章介绍一个 MySE 系统中使用的 IoC 容器——Hivemind，通过对这个 Java 框架的介绍来阐述了一些设计模式与软件工程的知识点。

编 者

目　录

第 1 章　网页数据挖掘

1.1　网页数据挖掘定义

数据挖掘(Data Mining，DM)，是从存放在数据库、数据仓库或其他信息库中的大量数据中提取或“挖掘”有趣知识的过程。随着网络的不断发展，因特网目前已成为一个巨大的、分布广泛的和全球性的信息服务中心。从海量的网络信息中寻找有用的知识，早已成为人们的迫切需求。各种类似 Google、Baidu 等的搜索引擎也层出不穷，Web 数据挖掘的应用在现实中不断体现。

Web 数据挖掘建立在对大量的网络数据进行分析的基础上，采用相应的数据挖掘算法，在具体的应用模型上进行数据的提取、数据筛选、数据转换、数据挖掘和模式分析，最后做出归纳性的推理、预测客户的个性化行为以及用户习惯，从而帮助决策和管理，减少决策的风险。

Web 数据挖掘涉及多个领域，除数据挖掘外，还涉及计算机网络、数据库与数据仓储、人工智能、信息检索、可视化、自然语言理解等技术。

1.2　Web 数据挖掘面临的问题

Web 的巨大、分布广泛和内容多样使得目前的 Web 数据挖掘面临着众多问题和挑战。首先，对有效的数据仓库和数据挖掘来说，Web 上的数据过于庞大。而且，Web 上的数据具有极强的动态性，不仅数量增长快而且更新十分迅速。但是面对如此大量的 Web 信息，却有调查表明：99%的 Web 信息对于 99%的用户是无用的。这样看来，面对网络上形形色色的用户群体，许多由 Web 搜索引擎所检索到的资料将会被淹没。

另外，由于 Web 页面缺乏统一的结构，其结构又比任何传统的文本文档都要复杂，所以要实现基于 Web 的数据挖掘和信息检索在目前来说是非常具有挑战性的。

Web 数据挖掘是一项具有挑战性的课题。它实现对 Web 存取模式、Web 结构和规则以及动态的 Web 内容的查找。

1.3　Web 数据挖掘的分类

一般来说，Web 数据挖掘可分为四类： Web 内容挖掘、Web 结构挖掘、Web

使用记录挖掘和 Web 用户性质挖掘。其中，Web 内容挖掘、Web 结构挖掘和 Web 使用记录挖掘是 Web 1.0 时代就已经有了的，而 Web 用户性质挖掘则是伴随着 Web 2.0 的出现而出现的。

Web 内容挖掘主要包括文本挖掘和多媒体挖掘两类，其对象包括文本、图像、音频、视频、多媒体和其他各种类型的数据。这些数据一般由非结构化的数据(如文本)、半结构化的数据(如 HTML 文档)和结构化的数据(如表格)构成。对非结构化文本进行的 Web 数据挖掘，称为文本数据挖掘或文本挖掘，是 Web 数据挖掘中比较重要的技术领域。Web 数据挖掘中另一个比较重要的技术领域是 Web 多媒体数据挖掘。

目前，关于 Web 内容挖掘的研究大体以 Web 文本内容挖掘为主。Web 内容挖掘一般从资源查找和数据库两个方面进行研究。

从资源查找的方面来看，Web 内容挖掘的任务是从用户的角度出发，怎样提高信息质量和帮助用户过滤信息，主要是对非结构化文档和半结构化文档的挖掘。非结构化文档主要指 Web 上的自由文本，如小说、新闻等。Web 上的半结构化文档挖掘是指在加入了 HTML、超链接等附加结构的信息上进行挖掘，其应用包括超链接文本的分类、聚类、发现文档之间的关系、提出半结构化文档中的模式和规则等。

从数据库的方面来看，进行 Web 内容挖掘主要是试图建立 Web 站点的数据模型并加以集成，以支持复杂查询，而不只是简单的基于关键词的搜索。这要通过找到 Web 文档的模式、建立 Web 知识库来实现。

对文本数据进行挖掘的文档分类和模型质量评价方法与传统的数据挖掘方法类似，分类算法主要应用朴素贝叶斯(Naive Bayes Classifier)。对模型的质量评价主要有分类的正确率(Classification Accuracy)、准确率(Precision)和信息估值(Information Score)。

Web 多媒体数据挖掘是指从多媒体数据库中提取隐藏的知识、多媒体数据关联或者是其他没有直接储存在多媒体数据库中的模式。Web 多媒体数据挖掘包括对图像、视频和声音的挖掘。Web 多媒体数据挖掘首先进行特征提取，然后再应用传统的数据挖掘方法进行进一步的信息挖掘。对网页中的多媒体数据进行特征的提取，应充分利用 HTML 的标签信息。

Web 数据挖掘是当今世界上的热门研究领域，其研究具有广阔的应用前景和巨大的现实意义。目前，国内的 Web 数据挖掘尚处于学习、跟踪和探索阶段。Web 数据挖掘有许多问题有待于进一步的研究和深化。Web 2.0 的出现给 Web 数据挖掘提出了新的要求。基于 Web 2.0 的数据挖掘目前还处于起步阶段，它必将成为 Web 数据挖掘中很重要的一个研究领域。

1.4　网页数据的结构与特点

1.4.1　HTML 超文本标记语言

什么是 HTML 文件？HTML 的英文全称是 Hyper Text Markup Language，中文叫做“超文本标记语言”。和一般文本不同的是，一个 HTML 文件不仅包含文本内容，还包含一些 Tag，中文称“标记”。一个 HTML 文件的后缀名是.htm，或者是.html。用文本编辑器就可以编写 HTML 文件。

这就试写一个 HTML 文件吧！

打开你的 Notepad，新建一个文件，然后输入代码到这个新文件，最后将这个文件保存为 first.html，其中的代码如下：

```
<html>
    <head>
    <title>Title of page</title>
    </head>
    <body>
        This is my first homepage. <b>This text is bold</b>
    </body>
</html>
```

要浏览这个 first.html 文件，双击它，或者打开浏览器，在 File 菜单中选择 Open 命令，然后选择这个文件就行了。

1. 示例解释

这个文件的第一个 Tag 是<html>，这个 Tag 告诉浏览器这是 HTML 文件的头。文件的最后一个 Tag 是</html>，表示 HTML 文件到此结束。Tag 通常是成对出现的，比如<body></body>，起始的叫做 Opening Tag(起始标记)，结尾的就叫做 Closing Tag(结尾标记)。在<head>和</head>之间的内容，是 Head 信息。Head 信息是不显示出来的，在浏览器里看不到。但是这并不表示这些信息没有用处，比如，可以在 Head 信息里加上一些关键词，有助于搜索引擎搜索网页。在<title>和</title>之间的内容，是这个文件的标题。你可以在浏览器最顶端的标题栏看到这个标题。在<body>和</body>之间的信息，是正文。

HTML 文件看上去和一般文本类似，但是它比一般文本多了 Tag，比如<html>、等。在和之间的文字，用粗体表示。顾名思义，就是 bold 的意思。通过这些 Tag，可以告诉浏览器如何显示这个文件。

HTML 元素(HTML Element)用来标记文本，表示文本的内容。比如，body, p, title 就是 HTML 元素。 HTML 元素用 Tag 表示，Tag 以<开始，以>结束。目前

HTML 的 Tag 不区分大小写，比如，<HTML>和<html>其实是相同的。

2. HTML 元素(HTML Elements)的属性

HTML 元素可以拥有属性，属性可以扩展 HTML 元素的能力。比如，你可以使用一个 bgcolor 属性，使页面的背景色变成红色，这时的标记形如<body bgcolor="red">。再比如，你可以使用 border 属性，将一个表格设置成一个无边框的表格，这时的标记形如<table border="0">。

属性通常由属性名和值成对出现，形如 name="value"。上面例子中的 bgcolor, border 就是 name，red 和 0 就是 value。属性值一般用双引号标记起来。属性通常是附加给 HTML 的 Opening Tag，而不是 Closing Tag。

1.4.2 WML 无线标记语言

WML(Wireless Markup Language，无线标记语言)。这种描述语言同我们常听说的 HTML 语言同出一家，都属于 XML 语言这一大家族。WML 的语法跟 XML 一样，WML 是 XML 的子集。HTML 语言写出的文件，我们可以在 PC 机上用 IE 或是 NetScape 等浏览器进行阅读，而 WML 语言写出的文件则专门用来在手机等的一些无线终端显示屏上显示，供人们阅读，并且同样也可以向使用者提供人机交互界面，接受输入的查询等信息，然后向使用者返回他所想要获得的最终信息。

1. WML 文件结构

WML 由一组互相链接的卡片(CARD)组成。当移动电话访问一个 WML 页面的时候，页面的所有 CARD 都会从 WAP 服务器下载到设备里。CARD 之间的切换由电话内置的处理器处理，不需要再到服务器上取信息。CARD 可以包含文本、标记、链接、输入控制、任务(TASK)、图像等。CARD 之间可以互相链接。

文档的实体包含在<wml>和</wml>标记之间，文档里每个 CARD 又包含在<card>和</card>标记之间，实际的文字段落则包含在 和 标记之间。

下面是一个简单的例子：

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
   "http://www.wapforum.org/DTD/wml_1.1.xml">

<wml>
<card id="HELLO" title="HELLO">
  <p>
  Hello world!
  </p>
</card>
</wml>
```

显示结果如下：

```
------ HELLO ------
Hello World!
```

2. WML 字符集

WML 是 XML 的子集，继承了 XML 的字符集设置。WML 文档默认的字符集是 UTF-8。要显示中文，有两种办法。最简单的办法就是在文档头使用 encoding，即把第一行改为：

```
<?xml version="1.0" encoding="gb2312"?>
```

然而令人失望的是，有些手机和模拟器并不支持这种方法(将来会的)。所以目前第二种方法更普遍：不改变字符集设置，但是在写中文的时候用 UNICODE 代表中文字符，如：

```
<b> &#x901A;&#x8BAF;&#x5F55;</b>
```

代表：

```
通讯录
```

3. WML 元素：标记(Tag)和属性

WML 的主要内容是文本，由于标记会降低与手持设备的通讯速度，所以 WML 标准里仅仅使用了很少一部分标记。用于表格和图像的标记几乎都被排除了。

与 XML 一样，在 WML 语言中，所有元素都放在符号<和>中，并且包含一个开始标记、一个结束标记和一个内容标记，或者使用自身结束的控制标记。就像这样：

<tag>内容</tag>　例如： Hello World!

或<tag/>　例如：
 和 <go href="#done"/>

WML 同样支持在标记中标出属性。属性是标记的附加信息，与元素的内容不一样，它并不在屏幕上显示出来。属性通常在元素的开始标记后指定。如上面最后一个例子。

由于 WML 是 XML 的一种应用，因此所有的 WML 标记和属性都是区分大小写的(<wml>跟<WML>完全不同)，而且所有的标记都必须正确地结束。WML 要求属性的值必须放在双引号或单引号内，单引号可放在属性标记内或双引号内，字符亦可作为属性的值。

4. WML 注释

XML 支持这样的注释格式：<!--这句话你在手机上看不到-->

这些注释在浏览器中并不显示出来。WML 不支持嵌套元素注释。WML 外部引

用方式跟 HTML 相同，都用<href a="address.html">desc这种形式书写。

WML 是一种比较严格的语言，字符使用必须遵守相应的规则，这些基本规则主要包括以下几个方面：

(1) 大小写敏感。在 WML 中，无论是标签元素还是属性内容都是大小写敏感的，这一点继承了 XML 的严格特性，任何大小写错误都可能导致访问错误。

一般来说，WML 的所有标记、属性、规定和枚举及它们的可接收值必须小写，Card 的名字和变量可大写或小写，但它是区分大小写的。包括参数的名字和参数的数值都是大小写敏感的，例如 variable1、Variable1 和 vaRiable1 是不同的参数。

(2) 空格。对于连续的空字符，程序运行时只显示一个空格。属性名、等号(=)和值之间不能有空格。

(3) 标记。标记内属性的值必须使用双引号(")或单引号(')括起来。对于不成对出现的标记，必须在大于号(>)前加上右斜杠(/)，比如换行标记
必须写成
才正确。

(4) 不显示的内容。在 WML 中，不显示的字符主要包括换行符、回车符、空格和水平制表符，它们的 8 位十六进制内码分别是 10、13、32 及 9。

程序执行时，WML 将忽略所有多于一个以上的不显示字符，即 WML 会把一个或多个连续的换行、回车、水平制表符及空格转换成一个空格。

(5) 保留字符。这是 WML 的一些特殊字符，如小于号(<)、大于号(>)、单引号(')、双引号(")、和号(&)。如果需要在文本中显示这些字符，则在程序中必须转义。这种指定方式在 WML 中称为字符的实体(Entity)，比如&就是&amp 的实体，&lt 就是小于号(<)的实体。

1.5 网页数据挖掘的基本方法

网页数据挖掘的关键部分是信息提取，本节将主要介绍如何通过编写特定格式的表达式来表述网页信息提取的各种需求。

名词定义：

信息提取表达式(Information Extracting Expression，IEE)：用于定义如何提取页面中的信息的表达式。

表述格式：

在两行横线之间的内容是 IEE 的实际内容，比如：

```
--------------------------
匹配:    <title>[$filmTitle]</title>
--------------------------
```

上述 IEE 表达式的实际内容就是：

```
匹配：  <title>[$filmTitle]</title>
```

上下的两行横线起到分隔符的作用。

1.5.1　预备知识

1．正则表达式

什么是正则表达式？

“正则表达式”是一个描述一组字符串的模板。正则表达式是使用多种操作符来组合更小的表达式构建类似算术表达式。建立块的基本原则是正则表达式匹配一个单字符。多数字符，包括所有的字幕和数字，都是匹配它们自己的正则表达式。任何带有特殊含义的字符可以以反斜杠开头来进行引用。

2．正则表达式特殊字符

正则表达式可以跟随几个重复操作符(通配符)，正则表达式操作符如表 1.1 所示。

表 1.1　正则表达式操作符

操 作 符	效　果
.	匹配任何单个字符
?	之前的项目是可选的，匹配最多一次
*	匹配出现零次或者多次的先前项目
+	匹配一次或者多次先前项目
{N}	精确匹配 N 次先前的项目
{N,}	先前的项目匹配 N 或者更多次
{N,M}	先前的项目匹配至少 N 次，但是不多于 M 次
-	表示范围如果不是列表中最先或者最后或者一个范围的结束点
^	匹配行开始的空字符串；也表示不在列表范围内的字符
$	匹配行末的空字符串
\d	匹配数字
\s	匹配空格

两个正则表达式可以用中缀操作符“|”连接起来；作为结果的正则表达式匹配任何匹配字表达式的字符串。

1.5.2 变量模板匹配方法

假设我们要采集 http://www.mtime.com/movie/10057 这个页面上的信息。

1. 简单模式

先从最简单的做起，从页面中取出页面标题。在这个实际的页面中，标题是这样一段内容：

```
<title>
    阿金 The Stunt Woman(1996) - Mtime时光网
</title>
```

我们可以这样表达出信息提取的需求：

```
-----------------------------------------------
    匹配：  <title>[$filmtitle]</title>
-----------------------------------------------
```

上面这句话表达出了这样一个意思：我要找这个页面的标题，而标题的内容是<title>和</title>之间的内容，取到标题内容后，存进名为 $filmtitle 的变量。

2. 正则限制条件模式

假设某页面上有如下内容:

```
图片总共的页数：共10页。
```

如果我们想取出数字 10，可以这样表达：

```
-----------------------------------------------
匹配：  共[$pagenum(\d*?)]页
-----------------------------------------------
```

这样的话，就能把数字 10 取出来，并存放入变量 $pagenum。为什么要加上(\d*?)呢？加上这个表达式就是说，限制需要提取的文本必须是数字。如果没有这个限制的话，就会匹配到“图片总共的页数”这句话中的“的”字，于是 $pagenum 就变成“的”了，这是错的。

3. 正则模板模式

假设某页面上有如下内容：

```
1: <a href="xxx.com/song.mp3">菊花台</a>
2: <a href="xxx.com/song2.mp3">发如雪</a>
.....
<a href="xxx.com/zhoujielun.jsp">周杰伦</a>
```

我们试图用以下表达式：

```
-----------------------------------------------
    匹配：  <a href="[$url]">[$songname]</a>
-----------------------------------------------
```

提取出页面中的歌名和下载地址，但是上述表达式同样能匹配到

周杰伦

这一行，这并不是我们所希望的，观察页面后发现，歌名之前是有数字序号的，而歌手之前没有，于是，可以把 IEE 表达成：

```
-----------------------------------------------
    匹配：  \d+:  <a href="[$url]">[$songname]</a>
    修饰符：  正则模板模式
-----------------------------------------------
```

上述的\d+表明，我们需要匹配的模板的开始位置必须是一个或多个数字，上述的“修饰符：　正则模板模式”表明这是一个用“正则模板模式”处理的 IEE。上述 IEE 可以从页面中提取出如表 1.2 所示的信息。这正是我们需要的结构化信息。

表 1.2　从页面中提取出的信息

url	songname
xxx.com/song.mp3	菊花台
xxx.com/song2.mp3	发如雪

补充说明：“正则限制条件模式”与“正则模板模式”可以混合使用。

4．组匹配模式

考虑如下例子。

页面内容：

```
电影名：黄金甲
主演：<a href="zhoujielun.htm">周杰伦</a><a href="zhourunfa.htm">周润
发</a><a href="gongli.htm">巩俐</a>...更多

电影名：色戒
主演：<a href="zhoujielun.htm">梁朝伟</a><a href="zhoujielun.htm">汤
唯</a>...更多
```

我们希望把每个主演以及对应的链接提取出来，为了实现这个需求，IEE 表达式如下：

```
-----------------------------------------------
匹配：    '电影名：[$fileName]主演：[%actorInfo=<a
href="[$url]>[$actor]</a>%]...更多'
修饰符：'#组模式{& \s};'
-----------------------------------------------
```