Web信息采集研究进展

李盛韬 余智华 程学旗 白 硕
(中国科学院计算机技术研究所 北京100080)

A Survey on Web Crawling

LI Sheng-Tao YU Zhi-Hua CHENG Xue-Qi BAI Shuo
(Software Division, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100080)

Abstract As a basic component of search engine and a series of other services on Web, Web crawler is playing an important role. Roughly, a Web crawler is a program which automatically traverses the Web by downloading documents and following links from page to page. This article detailedly explains the principles and difficulties on the Web crawler, comprehensively argues several hot directions of Web crawler, and at last views the new direction of Web crawler.

Keywords Web crawling, Web gathering, Search engine, WWW, Agent

1. 简介

随着Internet/Intranet的迅速发展,网络正深刻地改变着我们的生活。而在网上发展最为迅猛的WWW(World Wide Web)技术,以其直观、方便的使用方式和丰富的表达能力,已逐渐成为Internet上最重要的信息发布和传输方式。然而,Web信息的急速膨胀,在给人们提供丰富的资源的同时,又使人们在对它们的有效使用方面面临一个巨大的挑战。为此,人们发展了以Web搜索引擎为主的检索服务,并且随着应用的深化和技术的发展,单纯的检索服务正在向信息转播、个人代理、个性化主动服务等领域全方位拓展。

作为提供这些服务系统的重要组成部分,Web信息采集正应用于搜索引擎、站点结构分析、页面有效性分析、Web图进化、内容安全检测、用户兴趣挖掘以及个性化信息获取等多种服务和研究中。随着Internet中信息的迅速膨胀和人们对各种服务质量要求的提高,Web信息采集的任务也越来越艰巨。可是,许多成功的和主流的采集系统,由于商业需要而将许多的技术细节匿而不宣,这对于Web信息采集的发展非常不利。尽管如此,国内外仍然对它进行了许多积极的研究。并且,随着应用的不断扩展和各种研究的不断增多,它现在已经成为一个较为独立的热门领域。

本文介绍主流Web信息采集的基本原理,并就两个实际的采集系统做了简要说明;给出了Web信息采集所面临的几大挑战和相应的技术手段;按照当前流行的几种采集思路,展现一些研究系统和流行系统最近的研究进展。最后,就Web信息采集的发展方向进行了展望。

2. Web信息采集的基本原理

Web信息采集(Web Crawling),主要是指通过Web页面之间的链接关系,从Web上自动地获取页面信息,并且随着链接不断向整个Web扩展的过程。实现这一过程主要是由Web信息采集器(Web Crawler)来完成的。Web Crawler也常称作Web Spider、Web Robot或Web Worm。粗略地说,它主要是指这样一个程序,从一个初始的URL集出发,将这些URL全部放入到一个有序的待采集队列里。而采集器从这个队列里按顺序取出URL,通过Web上的协议,获取URL所指向的页面,然后从这些已获取的页面中提取出新的URL,并将它们继续放入到待采集队列里,然后重复上面的过程,直到采集器根据自己的策略停止采集。对于有些采集器,到此就算完结了,而对于另一些采集器,它还要将采集到的页面数据和相关数据存储、索引并在此基础上对内容进行分析。

2.1 Web信息采集系统的基本结构

如图1所示,Web信息采集系统基本上可以划分为七个部分:URL处理器、协议处理器、重复内容检测器、URL提取器、Meta信息获取器、语义信息解析器和数据库,它们协调起来从Web上获取信息。图中的箭头表示数据走向。

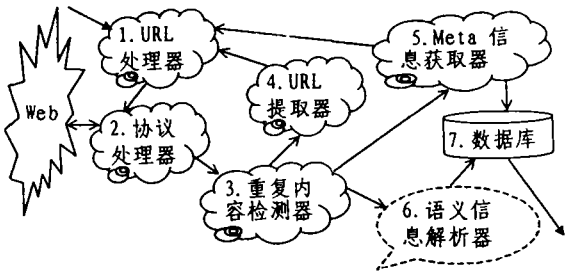

图1 Web信息采集系统基本结构

2.1.1 URL处理器 这个部件主要给待采集的URL排序,并根据一定的策略向协议处理器分配URL。按照采集系统规模的不同,URL可以是多个采集队列,也可以是一个URL Server。比如,天罗Web采集系统采用了多个采集队列,而Google采集系统则使用了URL Server,以达到更快的处理速度。URL处理器主要有三个数据来源:1)初始的种子

李盛韬 硕士研究生,主要研究方向:智能Agent,信息采集,信息检索,文本分类。余智华 博士研究生,项目主管,主要研究方向:信息检索与分类,个性化推送模型。程学旗 副研究员,主要研究方向:信息检索,机器翻译,数据挖掘。白 硕 博士生导师,研究员,主要研究方向:自然语言学,人工智能,网络安全。

URL 集，如图中的粗箭头所示；2)从 URL 提取器传输过来的 URL 集，它们是从已经采集到的页面中提取出来的；3)页面的 Meta、主题以及摘要等信息，来自 Meta 信息获取器，它们主要用来显示从 URL 提取器中传输过来的 URL 的重要性，为在这里排序提供依据。另外，URL 处理器还有一个任务就是 DNS 解析。

2.1.2 协议处理器 这个部件处于系统的底层，主要通过各种 Web 协议来完成数据的采集。一般来说协议包括 HTTP、FTP、Gopher 以及 BBS，也有些采集系统根据应用的需要采集 Web Chat、ICQ 等特殊信息。但从主流上看，仍以 HTTP 为主。下面简要说一下对 HTTP 协议页面采集的基本步骤：

1)按照页面 URL，抽出目标站点地址和端口号，若无端口号设为 HTTP 默认端口80。判断该站点的连接方式设置，若设为直接连接则与该地址和端口建立网络连接；若设为穿越 Proxy 连接则与指定的 Proxy 地址和端口建立网络连接。

2)若建立网络连接失败，说明该站点不可达，中止抓取该页面并将其抛弃；否则继续下一步获取指定页面。

3)由页面 URL 组装 HTTP 请求头，若该站点需要用户标识和口令则将其填入请求头中，发送请求到目标站点。若超过一定时间未收到应答消息则中止抓取该页面并将其抛弃；否则继续下一步骤分析应答消息。

4)分析应答头，判断返回的状态码：若状态码为2xx，返回正确页面，进入步骤5)；若状态码为301或302，表示页面被重定向，从应答头中提取出新的目标 URL，转入步骤3)；若返回其它状态码，说明页面连接失败，中止抓取该页面并将其抛弃。

5)从应答头中提取出日期、长度、页面类型等页面信息。若设置了页面抓取限制，进行必要的判断和过滤，抛弃不符合要求的页面。

6)读取页面的内容。对于长度较大的页面，采用分块读取再拼接的方法保证页面内容的完整。至此该页面的抓取完成。

2.1.3 重复内容检测器 Web 上存在着大量的镜像页面和内容，最近的研究表明，将近30%的页面是重复的。这极大地浪费了网络的带宽和影响了系统的效率。所以，重复内容检测变成了采集系统，特别是大型采集系统的重要组成部分。要采用的检测方法，根据系统的需要，从简单的段落匹配到复杂的相似度比较等中选择。

2.1.4 URL 提取器 对于采集到的页面，经过重复内容检测后，需要分析其中的链接，并对链接进行必要的转换，这些任务由 URL 提取器来完成。首先判别页面类型，对类型为“text、html、shtml 和 htm”等的页面进行分析链接。页面的类型可由应答头分析得出，有些 WWW 站点返回的应答信息格式不完整，此时须通过分析页面 URL 中的文件扩展名来判别页面类型。需要分析的标记包括〈a href＝……〉、〈area href＝……〉、〈base href＝……〉、〈frame src＝……〉、〈img src＝……〉、〈body background＝……〉和〈applet code＝……〉等。页面链接中给出的 URL 可以是多种格式的，可能是完整的包括协议、站点和路径的，也可能是省略了部分内容的，或者是一个相对路径。为处理方便，一般先将其规格化成统一的格式。

2.1.5 Meta 信息获取器 这里所要获取的内容包括已采集页面的 Meta 信息、页面的主题、页面的摘要等。获取它们的主要目的是力图在没有对页面内容语义信息进行理解的下，尽可能多地挖掘 meta、结构等的语义信息，来为从这个中提取出来的 URL 的好坏，给出一个度量。度量的结输到 URL 处理器，用于排序。

2.1.6 语义信息解析器 根据采集策略的不同，有些采器还有语义信息解析器。这里所说的语义信息解析就是指文本内容建立简单的索引。因为它在一定程度上挖掘了页面内容的语义，所以叫做语义信息解析器。对于一些大型的信息采集器，比如 Alta Vista，由于采集的信息量很大，对语义挖掘的深度要求较高，因此一般将页面语义挖掘与信息采集独立开来，而用专门的 Indexer 等部件进行处理。对于一些轻量级的采集系统，比如基于用户个性化的采集，因为采集的信息量不大(这样语义信息解析就不太影响采集效率)和采集过程中更需要语义信息制导，所以它们也常用到语义信息解析器。

2.1.7 数据库 经过重复内容检测后的页面数据、提取出来的 Meta 信息、主题和摘要等都要存入数据库，以备其他应用使用。比如，对于 Google 这样的搜索引擎，这个数据库中的内容将用于建立索引。如果系统有语义信息解析器，则解析出来的内容也存入数据库。由于数据较多，因此在存入数据库之前，数据一般要进行压缩。

下面就以两个实际的采集系统为例来具体说明，它们既有一般采集器的基本特点，也有自己的特色。

2.2 Mercator 信息采集器的基本结构和工作过程

Mercator 信息采集器是一个由康柏研究中心研制的面向整个 Web 的分布式多线程信息采集系统[11]。它的基本结构如下图所示，采集步骤是从1)到8)不断循环。步骤1)就是从多个线程共享的 URL Frontier 中移出绝对路径的 URL 来。绝对路径的 URL 中指明了这个 URL 采用什么方式下载。具体和协议相关接口的实现在 Protocol Modules 中。并且，用户可以通过设置文件来告诉系统装载哪些协议接口。

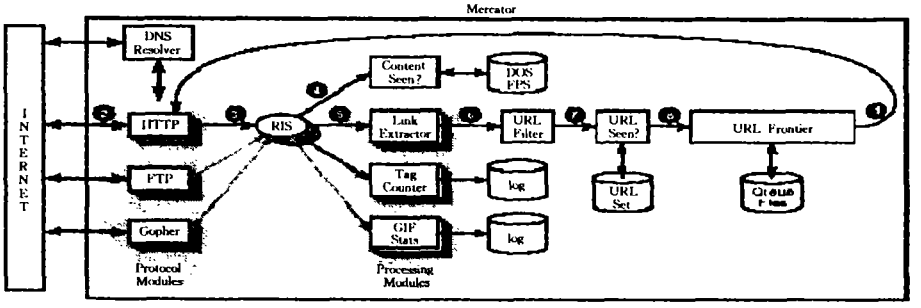

图2 Mercator 信息采集器结构

在步骤2)中，系统选择了相应的协议，通过了 DNS 解析并从 Web 上下载了页面，然后将页面放入3)RewindInput-

Stream(RIS)中,RIS 相当于一个缓存,能够多次快速地读内容。一旦文件被放进 RIS,这个工作线程就启动内容检测模块看是否此页面已经被采集过,这就是步骤4)。如果采集过,系统就抛弃此页并跳至步骤1)。

如果此页没有采集过,就进入步骤5)Processing Modules,在这里对页面进行初步的分析,比如提取标题、摘要和链接。缺省状况下,页面中的所有链接都被提取出来,并转换成绝对 URL。然后进行步骤6),也就是根据用户要求对 URL 进行过滤(Filtering)。如果 URL 通过了过滤器,则检查此 URL 是否已经在 URL 待采集库中(步骤7)。如果此 URL 没有,则将它加入到 URL Frontier 中,等着被选中进入下一轮循环(步骤8)。

2.3 天罗信息采集系统的基本结构和工作过程

天罗信息采集系统是在国家"863"计划下由中科院计算技术研究所软件研究室开发的智能导航系统的子系统。本采集系统最初的目标是面向整个 Web 的信息采集,随着 Web 服务向个性化主动服务等领域的拓展,本采集系统的后续版本在中科院计算所领域前沿青年基金资助下正在向基于主题的采集和个性化定制的采集方向发展。

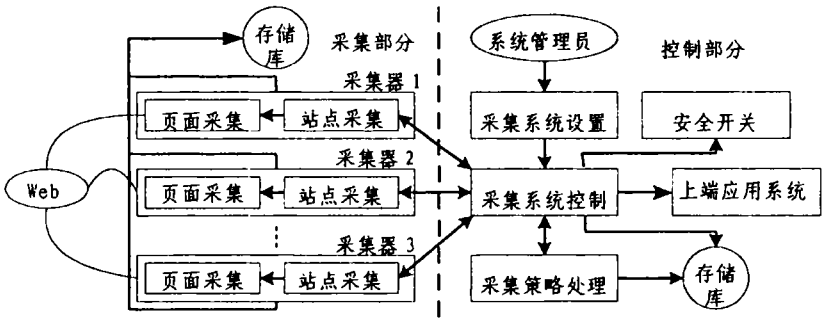

图3 天罗信息采集系统结构

如图3所示,天罗 Web 信息采集系统从功能上看可分为两个部分:采集器部分和控制部分,中间的竖立虚线将它们分开。采集器部分主要负责实际采集,它分为三个部分。1)站点采集。把整个 Web 以站点为单位划分成若干个连通子图是合乎人们的浏览习惯的,并且也是利于存储的。天罗 Web 信息采集系统的设计就是根据这一点,对 Web 上的页面以站点为单位进行采集。2)页面采集。尽管系统从粗粒度上看,采集是以站点为单位的,但是从细粒度上讲,每次只采集一页。这个部分考虑的重点就是对采集每页相关的协议的处理和实时网上异常的处理。3)存储库,主要存储采集到的数据、站点结构信息以及相关的有用信息。控制部分主要负责采集以外的协调、策略以及与应用的接口,它分为五个部分。1)采集系统设置,主要用于系统管理员对采集系统的控制,包括设置采集起点和采集策略。2)采集系统控制,这是采集系统最具有全局观念的一个子系统,它主要负责总体控制和其他各子系统之间的协调和连接,另外它还集中式地控制多个采集器并行。3)存储库,主要负责存储一致化处理后的各项数据以及在此基础上进行索引等处理的数据。4)采集策略处理,负责处理采集系统在理论上最难的一个部分--如何有效地采集和动态地刷新,另外,如何自动挖掘用户需求以及利用这些需求引导采集也是采集策略处理的一个主要问题。5)安全开关,在实际应用系统中,采集器往往直接和 Web 相连,而同时又与内部的应用服务器相连,如果不加安全处理,Web 对于应用服务器是非常危险的。为此,本采集系统设计了低成本高效率的安全开关。当与应用系统交换数据时,采集系统与 Web 断开;当在 Web 上采集数据时,采集系统与应用系统断开。这也是本采集系统的特色之一。图中的箭头详细地描述了数据流向。

为了提高采集的效率,天罗 Web 采集系统采用服务器(采集系统控制)/采集器的结构使采集系统具有很好的可扩展性。管理员可根据系统采集规模的变化动态地调整采集器的数量,在保证系统性能的前提下尽量减少系统开销,达到最佳的性能/价格比。而且在规模动态变化的过程中,系统能维持一致的管理和数据输出接口。

3. Web 信息采集面临的主要困难和相应的技术手段

粗看起来,采集的过程似乎比较简单,但实际上,它却存在着许多技术上和工程上的挑战。在分析这些挑战之前,先看一下 Web 的特点。

3.1 Web 的特点

Web 与传统的信息媒介相比,主要存在以下几个特点:1)信息容量的巨大性,在1998年7月,Web 上的静态文本大约有3亿5千万个,并且以每月600GB 的速度增长[13];2)Web 的动态性,每天 Web 中的内容和 Web 的结构都在变化着;3)Web 的异构性,Web 中包含的文件类型各式各样,包括图像、图片、声音、文本以及 Script 等;4)Web 页面的重复性,最近的研究表明,将近30%的页面是重复的;5)高链接性,平均每个页面有超过8个链接指向别的页面;5)多语种性,现在 Web 上的页面语种超过了100个。这为 Web 的有效采集,特别是为搜索引擎的采集提出了巨大的难题。

3.2 Web 采集面临的技术困难和相应手段

从技术角度看,挑战主要有以下几点:第一,Web 信息容量的巨大性使得采集器不可能采集到所有的 Web 页面,而且很多采集系统也没有足够大的空间来存放采集到的所有页面。如何提高采集的效率,即在单位时间内采集到尽可能多的高质量页面,是采集系统面临的一个难题,特别是对受限采集和基于主题的采集,更需要采用一定的策略对采集页面评分,以此来选择高质量页面优先采集。目前,有五种页面质量高低的计算方法:1)Similarity(根据页面和指导采集的问题之间的相似度);2)Backlink(根据这个页面在 Web 图中的入度大小);3)PageRank(根据指向它的所有页的平均权值之和,一页的平均权值定义为这页的权值除以这页的出度);4)Forwardlink(根据这个页面在 Web 这个图中的出度的大小);5)Location(根据这个页面的位置信息)。Cho 中对比了宽度优先方法、Backlink 方法和 Pagerank 方法[9],并根据试验比较

得出 PageRank 方法最好。这是因为 Pagerank 方法反映的是一种全局的页面质量分布，它能够较快地发现全局的高质量页面。当采集的页面范围较广时，它跟其它几种方法相比优势更加明显。

第二，并行性问题。页面的采集速度一直是影响采集器性能的重要原因。一方面，Web 中的页面数量非常庞大，另一方面，网络中的连接速度又非常缓慢，这客观上要求系统需要并行。然而，要并行又引入新的问题：1)重复性。多个不同的采集器或采集线程在同时采集的时候增加了重复页面；2)质量问题。单个系统能够根据算法采集到全局最优的页面。而如果并行，每个采集线程只能看到局部页面，它能够采集到的页面质量有所下降；3)通信带宽代价。为了并行，各个采集线程之间不可避免地要有一些通信。一般来说，采集系统采用两种模式来并行：局域网并行模式(所有的采集线程都运行在同一个局域网内，它们之间通过高速的内连接进行通信)和分布式并行模式(采集系统被分布在地域上较远的 Internet 上，通过网络进行远程通信)。在并行时，采集系统也可选用以下三种方式合作：1)独立方式。各个采集器之间独立地采集各自的页面，互不通信；2)动态分配方式。有一个中央协调器，动态地协调分配 URL 给各个采集器；3)静态分配方式。将 URL 按事先划分分配给各个采集器。对于静态分配方式，存在一种跨区链接(Inter-link，指一个采集器在提取链接时遇到的不属于自己采集范围的链接)。难题在于跨区链接并不一定能被它所属的采集器发现，如果本采集器采集了则可能重复采集，如果不采集则可能漏采。近期的研究工作针对这种情况比较了三种模式：1)防火墙模式。完全不采集 inter-link 页面；2)交叉模式。采集遇到的 inter-link 页面；3)交换模式。当采集到 inter-link，就将这些链接保存起来，积累到一定数量后传输给它们所属于的采集器进行采集。实验结果和我们的直觉一样：交换模式最好[6]。

第三，刷新问题。为了保持采集到的页面是最新的，采集系统不得不对已经采集过的页面进行周期性的更新。而随着 Web 的爆炸性膨胀，这个问题几乎变成了不可逾越的鸿沟。最近的一项报告显示，甚至流行的 Web 搜索引擎，页面刷新一次甚至持续数月[15]。同时，这份报告也显示，14%的搜索引擎提供的页面是无效的。Cho 试图用泊松过程来描述页面变化率，并研究和对比了三种页面刷新策略：固定顺序的刷新，随机刷新和纯随机刷新策略。直觉上，更多的刷新应该分配给那些更新快的页面。但研究表明，用较高的频率刷新更新快的页面并不一定是明智之举，对各种页面采用相同的刷新周期效果更好，而效率最佳的点在这两种刷新策略中间。这是因为，过高频率的刷新更新快的页面，使得其它页面有较少的刷新机会，反而造成总体刷新质量下降[6]。

3.3 Web 采集面临的工程困难和相应手段

从工程角度看：第一，正如前面分析的，Web 中的信息是完全异构、混乱和无序的，文件格式各式各样，网络状况也随时间变化莫测。所有的这些不确定性因素，为系统实现带来了极大的困难。这就要求采集系统有完善的异常处理和故障恢复机制，充分考虑到实际网络环境中可能出现的各种情况。

第二，多线程与并行机制使系统变得非常复杂。在这种复杂的环境下，系统的许多瓶颈变得异常突出，并需要采用多种设计技巧来解决。比如说，对一个网站的采集不能过分集中，以防止造成网站负担过重，Google 中的采集页面就是同时采集多个站点，以分散单位时间里每个站点的负担。在 Google 中，系统为每一个采集器都配了一个 DNS 缓存服务器，以加快 DNS 解析的速度。

4. 主要采集技术及发展方向

目前，对采集技术进行分类并没有一个统一的标准，但根据目前国际上一些流行的看法，Web 信息采集基本上可分为以下几种：基于整个 Web 的信息采集(Scalable Web Crawling)，增量式 Web 信息采集(Incremental Web Crawling)，基于主题的 Web 信息采集(Focused Web Crawling)，基于用户个性化的 Web 信息采集(Customized Web Crawling)，基于 Agent 的信息采集(Agent Based Web Crawling)，迁移的信息采集(Relocatable Web Crawling)，基于元搜索的信息采集(Metasearch Web Crawling)。实际的采集系统往往是几个采集技术相结合实现的。下面分别予以介绍。

4.1 基于整个 Web 的信息采集

这种信息采集在国外也常叫做 Scalable Web Crawling。主要是指目标为从一些种子 URL 扩充到整个 Web 的信息采集。这种信息采集主要是作为门户站点搜索引擎和大型的 Web 服务提供商的数据收集部分，由于商业原因，这部分的技术细节很少被公布出来。对于这类信息采集来说，由于采集的范围和数量都非常巨大，因此对采集速度和存储空间要求很高，为此对内存和磁盘的使用效率要求很高；由于目标是采集整个 Web，因此和下面要介绍的基于主题和用户个性化的信息采集相比，对采集页面的顺序要求相对较低，但考虑到高质量页面应用更多的状况，仍需要首先采集和刷新质量较高的页面；由于待刷新的页面太多，尽管并行很多的采集器，但仍需数周乃至数月的时间来刷新一次，而且，随着并行的采集器数量的增加，整个系统能力的提高越来越小，稳定性却越来越低；由于这类信息采集一般并行的采集器数量较多，因此要花较多的精力在 URL 分配、重复 URL 消除和全局 URL 质量计算上。总的来说，由这类 Web 信息采集构建的搜索引擎，适合搜索广泛的话题，并且几乎每一个搜索词都能搜索出一些相关结果来，所以这类信息采集有较强的应用需求，目前在实际应用中占较为主流的地位。下面通过简要分析三个实例 Google[3,9]、Mercator[11]和 Internet Archive[4]来进一步说明这类信息采集。

Google Crawler 是一个分布式的基于整个 Web 的采集器，它的主要研究工作在美国 Stanford 大学完成。Google 大部分是用 C/C++写的，整个结构设计力争避免磁盘访问时间。Google 并没有采用多线程技术，而是采用异步 I/O 管理事件来实现并行(一般的并行系统都采用这两种技术中的一种)。Google 有一个专门的 URL Server 来为并行的多个采集器维护 URL 队列。为了保持高速的获取页面，每个采集器一次同时打开大约300个连接。在使用4个采集器时，系统的峰值速度大约是每秒100页，这相当于每秒大约获取600k 的数据。由于系统的主要压力在 DNS 解析上，Google 为每一个采集器分配了一个 DNS Cache，这样不需要在每次采集页面时都做一次 DNS 解析。在每个采集器同时打开的近300个连接中，每一个连接都处于包括 DNS 解析、连接主机、发送请求和收到回复等状态之一。Google 使用许多算法对系统性能进行优化，最著名的就是 PageRank 算法，其思想是每页的权值为指向它的所有页的平均权值之和(一页的平均权值定义为这页的权值除以这页的出度)，这个权值用来评估所有采集页面的重要程度，它是一个递归定义。Google 也使用下面方法来分

散工作负载。首先，采集器将待采集的 URL 在根据 URL 所在站点服务器 IP 地址进行哈希函数计算以后放到自己的500个待采集队列中。这样，从同一站点服务器来的 URL 将被分配到同一个队列中。然后，采集器按一定的顺序从每个队列的队首读出一个 URL，这样从一个 IP 地址读出的两个 URL 之间至少隔499个其它地址的 URL。当采集器通过异步 I/O 而同时打开300个连接时，每个连接都来自不同的站点服务器，这样就有效地避免了由于目标站点服务器通信慢而带来的低效率以及对目标站点服务器造成的过高负载。Google 还选用了 zlib 压缩格式压缩采集来的页面数据，这是在时间和空间代价权衡后选择的。

康柏系统研究中心研究实现了 Mercator Web Crawler。与 Google 不同，它主要是使用 Java 实现的，在并行机制上，它则采用了多线程技术。一般情况下，每个采集器能够启动数百个线程。Mercator 也有一个单独的 URL 处理器(URL Frontier)，用于收集和合并从采集到的页面中提取出来的 URL。它有一个 RIS(RewindInputStream)部件，用于负责预览待采集页面以判断是否已经有相同内容的页面被采集过。Mercator 的另一大特点就是可扩展性(Extensible)，它允许用户根据实际的采集需要简单方便地插入模块代码，例如添加一种新的采集协议。设计者声称，在两个533MHz 的处理器、2G 内存和118G 硬盘的环境下，他们采集的速度是每秒112个文件，也就是大约每秒1682k 数据。另外，它有较低的404错误和8.5%的文件重复率。

Internet Archive 使用异步 I/O 技术来并行采集整个 Web，它的目标就是为整个 Web 存档。为此，每个采集器被分配64个站点，同时每个站点的所有页面都被分配给同一个采集器。在提取链接时，如果提取出的链接仍属于这个采集器，就将这个链接放入相应的待采集队列里，否则将它存在 log 文件中，积累一段时间后，再将 log 文件中的链接合并后按所属站点传输给相应的采集器。

4.2 增量式 Web 信息采集

这种信息采集在国外也常叫做 Incremental Web Crawling。传统上，Web 采集器根据自己的需要采集足量的信息后停止采集，当一段时间后这些数据过时时，它会重新采集一遍来代替原有的采集信息。这种采集器称作周期性 Web 采集器(Periodic Web Crawler)。而另外一种方法，对待旧的页面采用增量式更新，也就是说，采集器在需要的时候采集新产生的或者已经发生变化了的页面，而对于没有变化的页面不进行采集。理想状况中，已采集到的信息应该和 Web 中的信息是一致的，然而实际上 Web 的动态性、异构性和复杂决定了采集到的信息在相当短的时间内就可能过时，那种理想中的一致性是不可能实现的。所以我们能够做的就是尽量逼近这种一致性。由于和周期性信息采集相比，增量式信息采集能极大地减小数据的采集量进而极大地减小采集的时空开销，因此它成为实际系统的首选和研究热点。前面所说的 Google、Mercator 和 Internet Archive 都是增量式信息采集系统。但是，增量式信息采集在减小时空开销的同时，却增加了算法的复杂性和难度，比如说如何判断某个页团是否变化。同时，为了进一步提高增量式信息采集的效率，又面临新的难题，例如如何根据页面的变化快慢分配系统的采集能力。在最近的一项实验中[7]，随机选择了270个站点(包括132个.com 站点，78个.edu 站点，30个.net&.org 站点和30个.gov 站点)并下载了72000个页面，发现超过40%的.com 页面每天变化，.net 和.org 变化适中，而.edu 和.gov 变化最为缓慢。IBM 设计完成的信息采集器 WebFountain 是一个典型的增量式系统[10]。它采用了一个优化模型来控制采集策略。这个模型没有对 Web 页面变化的统计行为做任何假设，而是采用了一种适应性的方法，根据先前采集周期里采集到的结果的实际变化率进行调整。作者也提到为更新频率较快的页面提高刷新频率。

4.3 基于主题的 Web 信息采集

这种信息采集器在国外叫做 Focused Crawler，是指选择性地搜寻那些与预先定义好的主题集相关页面的采集器。主题一般可以是关键词，也可以是样本文件。和基于整个 Web 的信息采集器相比，基于主题的 Web 信息采集器并不采集那些与主题无关的页面，所以极大地节省了硬件和网络资源，保存的页面也由于数量少而更新快，比较接近当前的 Web。但它的问题也是显而易见的，如何定义有实际意义的主题、如何在采集时判定页面与主题相关以及如何控制关于主题的查全率等。对于选择主题来说，并不是每个词都能作为主题(比如很多动词)，也不是每个词都适合作为主题(各个主题词要内容不相交地划分全体知识)，一般认为选用类似于 Yahoo 这样的完整分类体系对全体知识进行划分较好，各个主题采集器只采集其中的一个或几个主题，各个不同的主题采集器之间还可以互相通信进行协作(比如，自己如果采集到其它主题的 URL，推荐给该主题的采集器)。和基于整个 Web 的采集器相比，基于主题的 Web 信息采集器的最大不同就是要对提取出来的 URL 加一个主题相关性判断。现在较常用的相似性判断方法是向量空间模型 VSM，即将主题和待比较 URL 转化成向量后通过余旋夹角公式计算其相似度。为了较准确地计算此值，主题一般要用若干个样本文件刻画而不仅仅是一个词，URL 则用包含此 URL 的页面、URL 前后的信息以及 URL 本身的信息三项内容来尽可能准确地预测此 URL 所指向页面的内容。基于主题的 Web 信息采集的一个目标就是要提供在本主题内比基于整个 Web 的信息采集好而全的页面，对于查全率问题有以下讨论。由于已经采集的页面少，发现隐含页面的能力较弱，这就需要各个领域采集器进行协作(其它采集器传输它们认为属于本主题的 URL 和页面给本采集器)；在计算相似性的时候，很可能会因为误判而漏掉一些属于本主题的 URL，解决的方法是放宽阈值限制，增加无关误判，减小相关误判，然后在采集到后进行页面相似性比较去除无关页面。一般的主题采集器为领域搜索引擎采集数据，也有些主题采集系统根据用户的主题词在 Web 上采集后直接呈现给用户，对于这类主题采集系统，相似性判断要求很高，一般阈值设置的都比较高。目前基于主题的 Web 信息采集的研究比较热门。下面分别介绍几个流行的系统和方法。

印度理工大学(IIT)和 IBM 研究中心的研究人员开发了一个典型的基于主题的 Web 信息采集器[5]。它的主题集是用样本文件来描述的。为了达到采集时主题制导的目的，设计者设计了两个文本挖掘的部件来指导采集。一个是分类器(Classifier)，用于评价采集文本是否与主题相关。另一个是精炼器(Distiller)，用于识别能够在较少的链接内就连接到大量相关页面的超文本节点。采集系统首先保存一个经典的主题分类(例如 Yahoo 的主题分类)，并且为每一个主题分类都保存若干个内容样本，用于详细地刻画这一类主题。用户在使用本采集器搜索与主题相关的页面时，必须在系统的主题分类树中先选择一个主题，用于指导采集。为了使采集的效果更佳，也可提供非常相关的文本样本或者 URL。由于要选择和

剪枝，采集速度并不太快，在双333MHz PII CPU，256M 内存 SCSI 硬盘下，每个采集器的采集速度为每小时6000页。

Aggarwal 则提出了一种针对两个假设的基于主题的 Web 信息采集方法[1]：1)Linkage Locality，即被相关于某一主题的页面链接到的页面趋向于拥有同一主题。2)Sibling Locality，对于某个链接到某主题的页面，它所链接到的其它页面也趋向于拥有这个主题。这样，在采集器接到一个主题采集请求命令后，它就从自己保存的关于这个主题的起点出发，按照两个假设蔓延，并利用指向备选页面中的 URL 结构以及其他一些 meta 信息使用统计学习的方法进行修剪，使采集的页面很快接近主题。

Web 上80%的内容是动态产生的，并且呈增长趋势[25]，而这些内容却几乎没有被采集下来。美国斯坦福大学的 Hidden Web Exposer Project 就是要建立一个采集这些动态页面的采集器[22]。因为很多隐式页面要通过填写表单等人工手段才能获取，并且基于不同的主题所填的内容和方式也不相同，所以采集器在采集之前需要人工辅助来事先填好领域信息，然后进行基于主题的采集。尽管主题信息的填写工作较繁琐，但同一主题的信息结构较相似，只要用户填写一次基本上就可以实现自动采集了(还需要用户进行少量干预)。

Menczer 则评价了三种关于基于主题采集的策略[19]：1) Best first Crawler(通过计算链接所在页面与主题的相似度来得到采集优先级)；2)PageRank(通过每25页计算一遍 PageRank 值来得到采集优先级，PageRank 值计算方法前面已经说过)；3)InfoSpiders(通过链接周围的文字，利用神经网络和遗传算法来得到采集优先级)。经过试验比较，作者发现，Bestfirst 方法最好，InfoSpiders 方法次之，PageRank 算法最差。一向被给予高度评价的 PageRank 算法之所以表现不佳，作者认为是它选出的高质量页面是基于广泛主题的，而对于特定主题来说页面的质量可能就不高了。

4.4 基于用户个性化的 Web 信息采集

不同的用户对一个搜索引擎提交同一个检索词，他们期望的返回结果是不同的，然而搜索引擎却只能返回相同的检索结果，这显然不能完全满足用户的需要。也就是说，提供给用户的服务，应该考虑到用户个性化的需求。为此，采集系统的设计者把目光投向了基于用户个性化的 Web 信息采集(Customized Web Crawling)。这是一种轻量级的采集系统，它的目标就是通过用户兴趣制导或与用户交互等灵活手段来采集信息。系统根据实际需要可以直接把采集结果提供给用户，也可以先存储起来等到以后再提供。这种个性化信息一般有两个来源，第一个是用户手工在系统提供的个性化设置页面里设置，这里主要考虑的问题是如何全面灵活简单地提供这种设置，使得用户的各种喜好都能够表达。第二个是系统自动获取，通过跟踪用户的浏览习惯和兴趣等。SPHINX 是一个 Java 工具包组成的环境交互式信息采集器[20]。它是一个能够迁移到站点服务器进行采集的个性化采集系统。用户的个性化设置嵌在工作台里，并且针对指定的站点进行个性化采集。文[12]中也介绍了一种基于 Web 上报纸新闻的个性化 Web 采集。这是个性化和主题采集应用结合的一个实例。

4.5 基于 Agent 的信息采集

随着智能 Agent 技术的发展，Agent 与信息采集相结合的技术也逐渐热门起来，这种采集技术叫做 Agent Based Crawling。智能 Agent 系统是指处以一定环境下包装的计算机系统，为了实现设计目的，它能够在该环境下灵活地自主地活动。它除了具有自治性(Agent 运行时不直接由人或其它东西控制，它对自己的行为和内部状态有一定的控制权)、社会能力(多个 Agent 体之间信息交换和协作)、反应能力(对环境的感知和影响)和自发行为(Agent 的行为是自主的)，还具有一般人类所有的知识、信念、意图和承诺等心智状态，这使得智能 Agent 系统具有人类的社会智能。它的这些特点使得它在面临诸如基于主题的采集和用户个性化的采集时，和传统方法比起来，更具方便灵活和适应力强等优势。比如说在基于用户个性化的采集中，它能像人一样感知用户的兴趣变化，并根据实际情况自主地迅速地灵活地智能地调整采集策略。因此，基于 Agent 的信息采集技术已经被用来进行 Web 上的资源发现、辅助浏览和信息采集。

美国的爱荷华大学进行的 ARACHNID 研究项目就是这方面的典型代表。它主要通过模拟一个生态系统的发展和演变来设计 Web 信息采集器 InfoSpiders[17,18]。系统的目标是从用户的角度在网上搜索最有效的页面。它的搜索过程是用户输入一个搜索要求，然后它根据搜索要求到网上去搜索并将结果返回给用户。它的采集原理可以看下面这个例子。以一个用户的书签或者其他表明用户兴趣的文件作为采集起点，通过分析这些起点周围的小区域和链接关系来发现新的要采集的页面。它通过对采集到的页面是否真的跟采集前的相关性预期相符，来增加和减少能量，当能量很高时，还可以生出新的孩子(新的子树)，而当能量过低时，它就死亡。当然，上面的用户兴趣也可以通过机器学习和相关反馈的方法进行调整。因为它是临时到网上去搜索，而不是像一般搜索引擎一样事先采集好了并完成索引到时直接匹配，所以尽管它的搜索精确度甚至更好，它的速度却是比较慢的。它的另一个好处是杜绝了过期页面。就目前的情况而言，取代门户搜索引擎(如 Google)是不现实的，所以他们选择的立足点是作为门户搜索引擎的有效补充。

美国麻省理工学院设计的系统 Amalthaea 是一个采用 Agent 技术设计的基于用户个性化需求的元信息采集器[21]。系统采用 Information Filtering agent 和 Information Discovery agent 来实施采集，前者的任务是挖掘用户个性化信息，后者的任务是根据用户要求到 Web 上进行信息搜索。这两个 Agent 的信息来源主要有下面三个方面：其它搜索引擎的搜索结果、根据这些搜索结果发现的新的种子页面和对可能变化的个性化 URL 的监控。这个系统的结构主要分为五层(从上到下为)：用户及其反馈、个性化 Web 浏览界面、信息过滤主体、信息搜索主体、分布式信息源。

美国麻省理工学院的另一个系统 Letizia 是利用 Agent 来辅助用户浏览 Web 页面的辅助工具[16]。当用户通过一个浏览器(比如说 Netscape)浏览页面时，Agent 就自动跟踪了用户的浏览行为，用启发式方法来估计用户的兴趣，并根据用户当前所在的位置，从网上采集一些满足用户当前感兴趣的页面推荐给用户。系统采用限制性宽度优先的方法，对用户最近浏览的兴趣页面向周围扩展。用户可以遵从这些推荐，也可以按着自己的方式浏览，而同时 Agent 不停地根据新的变化采集和推荐，推荐的内容紧跟当前用户的浏览页面。

美国 Stanford 大学研究了一种基于学习 Agent 的主题信息采集系统[2]。它使用向量空间模型 VSM 和 TF * IDF 来给发现的文本评分排序，并使用多种机器学习策略和用户反馈来修改启发式的搜索。

4.6 迁移的信息采集

这种信息采集器也叫 Relocatable Web Crawler。在采集时，它并不像其他采集器在本地向 Web 站点服务器发页面请求，而是将自己上载到它所要采集的服务器中，在当地进行采集，并将采集结果压缩后，回传到本地。这样做的一个明显优点是大量地节省了 Web 资源，特别是当这个采集器是基于用户个性化或者基于主题的采集器，大量的剪裁工作将在被采集对象的服务器上完成。但明显的一个不利是采集器可能并不被被采集对象所信任，因为这样被采集站点会由于给访问者权限太大而易遭到病毒攻击。解决的办法是建立一种信任机制，采集器由权威的信任机构评估并授权。还有另一种方法，采集器先迁移到离被采集站点很近的地方实施采集，这种方法是迁移到被采集站点方法和不迁移方法的折衷。SPHINX 信息采集器就是这种思路的尝试[20]。

4.7 基于元搜索的信息采集

元搜索引擎(Metasearch)的研究一直是搜索引擎研究的一个热点。它是这样一种搜索引擎系统，对用户提交的查询请求通过多个领域或门户搜索引擎搜索，并将结果整合后以统一的界面提交给用户。一般元搜索引擎并不保存 Web 页面的索引文件，但对于一些复杂的元搜索引擎，它要保存为它服务的每个搜索引擎的信息特征，以便能够在用户查询到来后做出好的搜索引擎选择。作为搜索引擎先头部队的信息采集器，在元搜索引擎中有相当的退化，但仍为 Web 采集的一个方向，叫做基于元搜索的信息采集(MetaCrawler)。元搜索引擎主要有以下几个好处：1)它帮助用户选择更适合用户查询领域的搜索引擎，当然，用户也可以指定若干个搜索引擎。2)它的返回结果比单个搜索引擎更全面，增加了整个信息索引的范围。3)在页面失效问题上，它可以通过合并领域性搜索引擎的结果来优于综合性搜索引擎(领域性搜索引擎页面失效问题优于综合性搜索引擎)。4)元搜索引擎的开发有更低的软硬件开销。

美国纽约的 Binghamton 大学的研究者围绕一个元搜索引擎技术的难点——数据库选择问题进行了研究[26]，并提出了一个解决上述问题的新方法。数据库选择主要是指根据用户提交的查询，选择合适的搜索引擎去搜索，因为要借用其它搜索引擎的索引数据，所以叫做数据库选择。他们的方法是：就每个有代表性的问题对大量的领域搜索引擎排序，这有点像建立索引时的倒排表。当一个检索词来了后，通过相似度比较选择一个最接近的代表性问题，进而确定了要选用的搜索引擎。

美国华盛顿大学在 Metasearch 方面的研究在文[24]有详细的说明。作者认为，大多数搜索引擎对于同一个查询要求返回的结果很不相同，质量也参差不齐。试验发现，使用单独一个搜索引擎错过大约77%的相关页面[23]。所以，他们力图在提高查全率的同时，也力争利用单个搜索引擎在某一领域的优势提高平均查准率。

结论 随着人们对 Web 服务的种类和质量要求越来越强烈，各种各样的信息采集系统也应运而生，并朝前不断发展。最初，人们希望能够设计出既大而全又质量好的信息采集系统(即基于整个 Web 的信息采集)，这显然是一个非常困难的问题，因为两方面都要求必然造成两方面都不能做得很好。人们经过不断的努力探索，从最初的 Web Worm 到现在的 Google，从基于词的语义信息理解到 Web 链接结构信息挖掘，发展到了今天已经取得了令人瞩目的进步，优秀的基于整个 Web 的采集器以及相关的搜索引擎，已经在很多方面为人们利用 Web 信息提供了大量帮助。然而，随着人们对 Web 服务的种类和质量要求越来越高，基于整个 Web 的信息采集也越来越显得力不从心，一方面它们不得不为越来越庞大的数据提高采集速度、增加存储空间、优化采集算法，而一方面又越来越不能满足用户对个性化数据的需求，人们需要寻找新的出路。目前采用的基于词的语义信息理解显然不能准确把握整个文章的语义，而要上升到对句子甚至段落信息的理解却还有待于自然语言理解的大发展，现在这一方面困难重重；基于已有结构信息的挖掘(例如 Google 的 Pagerank 算法)也已基本达到饱和，很难有新的算法达到较大突破；而对于纷乱的 Web 制定新的标准，减少不确定性以提高性能，这一方面的发展也不能寄予过高的期望；随着 Web 服务逐渐向基于主题以及用户个性化的方向迈进、Agent 的技术发展、迁移式思想的出现，Web 信息采集也找到了新的春天。单纯的为了检索的信息采集技术必将在信息转播、主题制导和 Agent 等思想的引导下，向着个性化主动信息采集服务方向全方位拓展，Web 信息采集技术必将有一个更加广阔的发展空间。

参考文献

1 Aggarwal, et al. Intelligent Crawling on the World Wide Web with Arbitrary Predicates. In: C. Aggarwal, F. Al-Garawi, P. Yu, eds. Proc. of the 10th Intl. WWW Conf. Hong Kong, May 2001

2 Balabanovic, Shoham. Learning Information Retrieval Agents: Experiments with Automated Web Browsing. In: M. Balabanovic, Y. Shoham, eds. Working Notes of the AAAI Spring Symposium on Information Gathering from Distributed, Heterogenous Environments, Stanford University, 1995

3 Brin, Page. The Anatomy of a Large-Scale Hypertextual Web Search Engine. In: S. Brin, L. Page, eds. Proc. of the Seventh Intl. World Wide Web Conf. Erisbane, Australia, April 1998

4 Burner M. Crawling towards Eternity: Building an archive of the World Wide Web. Web Techniques Magazine, 1997, 2(5)

5 Chakrabarti, et al. Focused Crawling: A New Approach to Topic-Specific Web Resource Discovery. In: S. Chakrabarti, M. van den Berg, B. Dom, eds. Proc. of the 8th Intl. WWW Conf., Toronto, Canada, May 1999

6 Cho. CRAWLING THE WEB: DISCOVERY AND MAINTENANCE OF LARGE-SCALE WEB DATA. Junghoo Cho. Ph. D dissertation 2001

7 Cho, et al. The Evolution of the Web and Implications for an Incremental Crawler. In: J. Cho, H. Garcia-Molina, eds. Proc. of 26th Intl. Conf. on Very Large Databases(VLDB), Sep. 2000

8 Cho, Molina. Synchronizing a database to improve freshness. In: Junghoo Cho, Hector Garcia-Molina, eds. Proc. of 2000 ACM Intl. Conf. on Management of Data(SIGMOD), May 2000

9 Cho, Molina, Page. Efficient Crawling Through URL Ordering. In: Junghoo Cho, Hector Garcia-Molina and Lawrence Page, eds. Proc. of the Seventh Intl. World Wide Web Conf. Toronto, Canada, May 1999

10 Edwards, et al. An Adaptive Model for Optimizing Performance of an Incremental Web Crawler. In: J. Edwards, K. McCurley, J. Tomlin, eds. Proc. of the 10th Intl. World Wide Web Conf. Hong Kong, May 2001

11 Heydon, Najork. Mercator: A Scalable, Extensible Web Crawler. A. Heydon and M. Najork. In World Wide Web Journal, Dec. 1999. 219～229

(下转第171页)

$$\hat{c}_i^{MAP} := \arg \max_{c_i^k \in \{c_1, c_2, c_3, c_4\}} p(c_i^k | c_{p(i)}^{k-1}) \quad (9)$$

其中，对 $P(c_i|c_{p(i)})$ 计算通过计算 $P(c_{p(i)}|c_i)$ 和下列贝叶斯公式完成：

$$P(c_i|c_{p(i)}) = \frac{P(c_{p(i)}|c_i)P(c_i)}{p(c_{p(i)})} \quad (10)$$

算法具体步骤为：从能可靠统计 ML 分割 c_{ML}^k 的初始尺度（设为尺度0）开始，用初始分割作为 c_{MAP}^0。在计算尺度 k（=1，2，…，J）时，用尺度 k－1上9个父类 $c_{MAP}^{k-1} = c_{p(i)}^{k-1}$ 作为其上下文，用公式7计算 $f(d_i^k|c_i^k)$，用公式10计算 $P(c_i^k|c_{p(i)}^{k-1})$ 后，选择能使 $f(c_i^k|d_i^k c_{p(i)}^{k-1})$ 最大的分类作为 c_{MAP}^k。

多尺度分类依赖算法从树根（最粗尺度）开始，一直延伸到最细尺度，每一步都综合考虑上一粗尺度上分类信息，从而提高分类准确性。在循环计算过程中，因为从大尺度到小尺度变化时，参数变化很小，所以采用大尺度已获得值作为初值时，算法的收敛速度很快。因小尺度概率分割可能不够精确，因此当尺度很细时，直接采用粗尺度上所获得参数，而不需要重新估算。

实际应用中，选择 j_0 的原则是确保在最粗尺度上对图像的分割比较可靠且有意义。一方面，当块尺寸太大时，可能包括多种图像域，其分类概率往往没有实际意义。另一方面，当块尺寸太小时，其分类概率计算到像素级也没有必要。因此，在考虑多尺度间依赖性时，只需考虑少数有意义的尺度，从而减少 HMT 尺寸、简化模型训练和分类判定所需计算。例如，可选开始分割尺度 $j_0 = 2^4$，相当于把一幅图像划分成一些 $2^4 \times 2^4$ 的子块 d_i^0，在每块上独立执行 HMT 概率计算；同时，如果选择最细尺度为 $j_1 = 2^2$，则只需要计算尺度为 2^4、2^3、2^2 时的小波系数、HMT 模型和分类概率。

小结 本文综合多种算法思想进行建模和计算，如小波域统计建模、直接概率计算、多尺度贝叶斯判定、多尺度分析等。描述了分割算法的主要思想和三个步骤：HMT 模型参数估算、多尺度概率计算、相关尺度分类概率融合。训练 HMT 模型通过人为方式，用特征明显的相似训练图像数据对每一种欲划分的图类（如双色文本、图片等）进行训练，获得各尺度上像素块分类 pdf 计算的 HMT 模型。基于 HMT 模型和概率计算公式估算各尺度上每个像素块的分类概率 $f(d_i^j|c_i^j)$。基于分类依赖关系融合相关尺度上像素块的分类概率，实现多尺度 MAP 分类。应用中需选择开始尺度0，使其在该尺度上对 d_i^0 分类能获得有意义的分类 c_i^0。

参 考 文 献

1 MRC. Mixed rarer content (MRC) mode. ITU Recommendation T. 44. 1999. 1～39

2 Unser M. Texture classification and segmentation using wavelet frames. IEEE Trans. Image Processing, 1995, 4(11): 1549～1560

3 Zhu Q S, Li B. An Algorithm of VBR coding for video transmission. See: M. H. Hamza. Computer Graphics & Imaging. Canada: IASTED/ACTA Press, 1998. 124～127

4 LeCun Y, Bottou L, Haffner P. DjVu: a Compression Method for Distributing Scanned Documents in Color over the Internet. AT&T Labs～Research. 1999

5 Fosgate C, Krim H, Irving W, Karl W. Multiscale segmentation and anomaly enhancement of SAR imagery. IEEE Trans. Image Processing, 1997, 6(1): 7～20

6 Wu C, Doerschuk P C. Tree approximations to Markov random fields. IEEE Trans. Pattern Anal. Machine Intell., 1995, 17(4): 391～402

7 Choi H, Baraniuk R G. Image segmentation using wavelet～domain classification. In: Proc. SPIE conf. Math. Modeling, Bayesian Estimation, Inverse Problems, USA, 1999, 3816: 306～320

8 Cheng H, Bouman C A. Trainable context model for multiscale segmentation. In: Proc. IEEE Int. Conf. Image Processing '98. Chicago, IL, 1998

9 Li Bo, Zhu Q S. The VBR Coding Algorithm Based on Wavelet Transform. Computer Engineering & Science, 1999, 21(1): 4～7

（上接第157页）

12 Kamba T, Bharat K, Albers M. The Krakatoa Chronicle - An Interactive, Personalized, Newspaper on the Web. In: Proc. of WWW 4, Boston, USA, Dec. 1995

13 Kahle B. Preserving the Internet, Scientific American, March 1997

14 Koster M. The Web Robots Pages. 1999

15 Lawrence S, Giles C L. Accessibility of information on the Web. Nature, 1999, 400(6740): 107～109

16 Letizia. An Agent That Assists Web Browsing. In: H. Lieberman, ed. Proc. of the Intl. Joint Conf. on AI, Montreal, Canada, Aug. 1995

17 Is Agent-Based Online Search Feasible?. In: F. Menzcer, ed. Working Notes of the AAAI Spring Symposium on Intelligent Agents in Cyberspace, Stanford, USA, March 1999

18 Adaptive Information Agents in Distributed Textual Environments. In: F. Menzcer, R. Belew, eds. Agents'98: Proc. of the 2nd Intl. Conf. on Autonomous Agents, Minneapolis, USA, May 1998

19 Evaluating Topic-Driven Web Crawlers. In: F. Menczer, G. Pant, P. Srinivasan, M. Ruiz, eds. Proc. of the 24th Annual Intl. ACM/SIGIR Conf. New Orleans, USA, 2001

20 Sphinx. A Framework for Creating Personal, Site-Specific Web Crawlers. In: R. Miller, K. Bharat, eds. Proc. of the 7th Intl. WWW Conf., Brisbane, Australia, April 1998

21 Amalthaea. Information Discovery and Filtering using a Multiagent Evolving Ecosystem. In: A. Moukas, ed. Proc. of the Conf. on Practical Applications of Intelligent Agents & Multi-Agent Technology, London, 1996.

22 Raghavan S, Garcia-Molina H. Crawling the Hidden Web: [Stanford Digital Libraries Technical Report]. 2000

23 Selberg E, Etzioni O. Multi-Service Search and Comparison Using the MetaCrawler. In: Proc. 4th World Wide Web Conf., Boston, MA USA, Dec. 1995

24 Selberg E, Etzioni O. The MetaCrawler Architecture for Resource Aggregation on the Web. IEEE Expert, 1997

25 Lawrence S, Lee C L. Giles. Searching the World Wide Web. Science, 1998, 280(5360): 98

26 Towards a Highly-Scalable and Effective Metasearch Engine. In: Zonghuan Wu, Weiyi Meng, Clement Yu, Zhuogang Li, eds. Proc. of the 10th Intl. World Wide Web Conf. Hong Kong, May 2001

27 朱淼良，邱瑜. 移动代理系统综述. 计算机研究与发展，2001(1)

28 马晓星，吕建. 分布式 Web 服务器技术综述. 计算机科学，2002(1)