分类号 密级

U D C 编号

中南大學

CENTRAL SOUTH UNIVERSITY

硕士学位论文

论文题目 垂直搜索引擎中 Web 信息抽取技术研究

学科、专业 计算机软件与理论

研究生姓名 胡国晴

导师姓名及
专业技术职务 李建华 教授

2008 年 7 月

摘　要

目前 Web 上的资源日益增多，为更有效地利用这些资源，近年来产生了垂直搜索引擎。它是面向专业或主题的搜索引擎，只采集与专业或主题相关的信息，这样就需要从 Web 页面等资源中抽取出特定的主题信息，本文的研究重点正是从 Web 页面中抽取与特定主题相关的信息。

针对目前 Web 信息抽取方法实现复杂等问题，设计了一种基于标签序列的 Web 页面主题信息抽取方法。该方法通过设定的策略和样本建立抽取规则，构建规则库，借助规则库实现对页面主题信息抽取，这样降低了处理 Web 页面过程的复杂性，并减少了页面处理时间。通过应用本方法抽取部分网站的手机参数页面，表明该方法召回率和准确率都比较高。

当需要抽取的 Web 页面结构发生变化而出现主题的新属性时，目前已有的方法建立的分装器并不能随着这种变化准确地发现主题的新属性。针对该问题，提出了一种基于可信度的 Web 页面主题新属性发现方法，通过对待抽取属性和已抽取属性的特点进行分析，引入可信度理论，通过一系列规则和证据，量化待抽取属性需要抽取的可信度，以判断待抽取属性是否为主题的新属性。并将其应用到部分网站页面手机参数主题属性发现中，实验证明该方法能够较为准确发现页面中主题的新属性。

最后设计了一种垂直搜索引擎原型系统, 主要完成了专业网络蜘蛛模块的详细设计，它综合了本文提出的 Web 页面主题信息抽取方法和 Web 页面主题新属性发现方法，以实现对页面主题信息的采集。

关键词 Web 信息抽取，垂直搜索引擎，主题信息，新属性

ABSTRACT

Information on the Web increases day by day, kinds of methods are proposed to make use of the information. The vertical search engine solves the problem partly. It's professional or topic-oriented search engine, only collecting professional or topic-related information and it extracts specific information from the Web. This paper is focused on extracting topic-related information from the Web pages.

At present, the implementation of Web information extraction is complex, a Web pages Topic-Information extraction method based on tag sequence has designed in the paper. The method sets a strategy, with the strategy and samples, we build the rule library and use the rule library to extraction Topic-Information from the Web pages, it reduces the complexity of processing Web pages and the processing time of the pages.Through using the method to extract phone parameters Web pages from some sites, we get a profect result on Recall and Precision in every site, it proves that our method is the application of feasibility.

For the problem that the Wrapper can't adapt to the change of structure for appearing Topic-Information new attribute on the Web pages, the paper presents a method to discover the Topic-Information new attributes on Web pages based on the credibility. Through analyzing the characteristic of the attributes which will be extracted and the attribute which has been extracted, introducing the theory of the credibility, it quantizes the credibility of what needs to be extracted of the attribute based on some rules and evidences, and decides whether the attribute is need to be found. Through using the method to find phone parameters attributes from some sites, it proves that our method can find the Topic-Information new attributes accurately.

At last, a vertical search engine prototype is designed and we mainly complete the special search spider module' design in detail. It integrates the Web topic information extraction method and new Web pages attributes discovery method which is proposed in this paper to collect the Topic-Information in the Web pages.

KEY WORDS Web Information Extraction, Vertical Search, Topic -Information, New Attribute

目 录

第一章 绪论

1.1 研究背景

近年来，随着信息时代的到来，互联网在我们的生活中的地位变得越来越重要，Web也获得迅速的发展。仅就中国而言，在07年1月的中国互联网发展报告中，全国网页总数估计为44.7亿个[1]，而这一总数在07年12月的调查中，中国网页总数已经达到84.7 亿个[2]。

互联网上的海量信息，如果能有效利用起来，对人类的发展必然能做出突出的贡献。然而随着互联网的迅速发展，当今社会并没有有效的管理互联网上的信息，从而造成目前网上信息混乱不堪，这极大的影响了用户快速、准确、完整的找到所需要的信息，搜索引擎正是为解决这种情况而诞生。

搜索引擎是指根据一定的策略、运用特定的计算机程序搜集互联网上的信息，在对信息进行组织和处理后，为用户提供检索服务的系统。目前搜索引擎已成为网民在汪洋中搜寻信息的工具，是互联网上不可或缺的工具和基础应用之一，在中国使用搜索引擎的比例已达到 72.4%，而美国已经达到 91%[2]。

根据搜索引擎定义，如何从互联网上采集信息成为搜索引擎的主要任务，Web信息抽取技术也就成为了搜索引擎的主要技术之一。而目前的搜索引擎主要是通用搜索引擎，如 Google、Baidu 等。但是在用户大部分查询条件下，通用搜索引擎返回的数据量太大，并且大部分查询的信息都与用户真正想要的信息无关；而且随着采集数据量的增大，这些信息的索引维护对系统的压力也不断的增大，这样必然损害用户查找信息的效率[3]。

针对通用搜索引擎的弊端，垂直搜索引擎的出现部分缓解了这种压力，垂直搜索引擎是针对某一个行业的专业搜索引擎，是通用搜索引擎的细分和延伸，它通过对网页库中某类专门信息进行一次整合，定向分字段抽取出需要的数据，并将该数据进行处理后以某种形式返回给用户[3]，它的基本原理与通用搜索引擎相同。

垂直搜索引擎和传统通用搜索引擎一个重要的区别是它对网页信息进行了结构化信息抽取，它属于 Web 信息抽取范畴。Web 信息抽取是通过对特定网页的结构和数据项特征进行分析，将网页预定义数据抽取出来，并为其增加一定的语义和模式信息[4]，如科技论文网页中可以抽取标题、作者、论文摘要、发布时间、关键字、正文等网页预定义数据。垂直搜索引擎是以结构化数据为最小单位，然后将这些数据存储到数据库中；而传统的通用搜索引擎是以网页或网页块为最小单位。所以 Web信息抽取的技术水平是垂直搜索引擎质量的重要技术指标，Web 信息抽取技术成为了垂直搜索引擎的关键技术之一。

1.2 研究现状

1.2.1 Web 信息抽取技术研究现状

目前，Web数据组织形式非常复杂，Web页面是它通常的表现形式，而组成页面的标签除了用于显示页面的数据外，并没有表达用户感兴趣的其他信息，如果没有人工或者先验知识的指导，电脑或程序很难发现用户感兴趣的信息。为了解决这个问题，Web信息抽取被提了出来，它属于信息抽取的一个分支。自20世纪90年代WWW诞生以来，国内外已经对Web信息抽取技术进行了多方面研究，本文根据抽取方法实现原理不同，具体分为以下四种。

1）基于自然语言理解的方法

基于自然语言理解的方法需要比较强的先验知识，需要事先通过训练样本建立自然语言语料库，并对语料库中词项等基本元素进行语义标注，在抽取时根据语料库建立需要抽取文档中短语或句子之间的关系，并归纳出抽取的规则，从而抽取所需要的信息[5]。

自然语言理解属于人工智能范畴，最先是用于机器翻译领域，在Web环境中，基于自然语言处理的方法通常应用于由语义信息构成的网页的抽取，如抽取公寓出租广告信息等。SRV[6]是由D.Freitag提出一种基于自然语言理解的信息抽取系统，在SRV中，通过事先给定一系列的训练集，对训练集中的样本进行手工标注，并通过自主学习生成抽取规则；E.Calif在他的博士论文中提出了一种信息抽取系统RAPIER [7]，它是一种面向自由文本的抽取数据的工具，利用了一系列机器学习的方法，通过关系学习生成抽取规则，以抽取感兴趣的信息；WHISK[8]是由S.Soderland提出的一种通过句法分析器和语义标注的抽取系统，它适合于各种形式的文本的抽取，同SRV一样，它也需要一系列的训练集，通过一个图形化接口对训练集样本中感兴趣的信息进行手工标注，并将被标注过的样本用来指导生成抽取规则。

2）基于HTML结构的方法

基于HTML结构的方法通过对Web页面结构进行分析，以实现对页面中相关信息的抽取。这种方法一般都将Web页面以网页结构树形式表示，通过对树分析实现抽取Web页面中信息。

目前该方法已经比较成熟，国内外已经对它进行了多方面研究，在国外William Cohen等人提出了一种基于分装器的信息集成系统WHIRL[9]，它先对Web页面生成页面结构树，并对树中的节点进行标记，并通过一系列启发式规则实现页面信息的抽取。YanYong Zhai等人提出了一种基于部分树(Partial Tree)的Web页面数据抽取算法[10]，该方法分为两个阶段，第一个阶段是机械学习标识样本页面中感兴趣的信息，第二个阶段是通过模式匹配发现页面中需要抽取的信息。DC Reis等人利用树编辑距

离(Tree Edit Distance)算法测量页面相似度，最后把相似度大于某个阈值的页面聚为同一类[11]，实现相似页面的信息抽取。在国内，李效东等人提出了一种基于DOM树的信息抽取方法[12]，该方法找出要抽取的信息在DOM树中路径, 并通过归纳学习生成抽取规则；陈琼等人提出了一种基于网页结构树的Web信息抽取方法[13]，他们通过在网页结构树中定位模式库中的待抽取信息，将对网页的信息抽取转化为对网页结构树的叶结点信息查找过程，进而实现对相关信息的抽取；微软亚洲研究院Peng Cai等人提出的VIPS算法[14]，该算法结合了DOM树和Web页面的视觉特征，它从一定程度上弥补了单独使用DOM树带来的缺陷。

3）基于Ontology的方法

基于Ontology的方法主要依赖一个完全的知识库。Ontology最早是一个哲学的范畴，随着人们的理解得不断完善，对Ontology的定义也出在不断的发展变化中，Ontology的目标是捕获相关领域的知识，提供对该领域知识的共同理解，确定该领域内共同认可的词汇，并从不同层次的形式化模式上给出这些词汇(术语)和词汇之间相互关系的明确定义。

该类信息抽取方法先对抽取页面的类型进行描述, 并根据待抽取信息特点设计出数据框架，并归纳出抽取规则，实现对页面信息的抽取，它对网页的结构依赖性比较低。主要代表有BYU(Brigham Yong University)信息抽取小组开发的信息抽取工具[15]，在BYU系统中，先由领域知识专家采用人工的方式书写某一主题的Ontology，然后根据Ontology中的值和关键字的描述信息产生抽取规则，从而抽取出感兴趣的信息；Ning Zhang等人提出了一种基于Ontology驱动的自适应抽取Web信息的方法[16]，该方法主要有两个步骤，一是构建一个主题相关的Ontology，二是设计了三种基于Ontology的自适应和自维护模板生成算法。在国内，张成洪等人利用正则表达式对文本处理的方便性，并结合Ontology实现对Web页面信息抽取[17]；廖乐健等人对线性模板表示做出了改进，提出了基于二测树结构的模板规则表示方法，并将Ontology与模板规则相结合，成功的应用到招聘广告信息抽取中[18]。

4）基于隐马尔可夫模型的信息抽取方法

基于隐马尔可夫模型(HMM)的信息抽取方法抽取信息查准率较高，但是需要较多的人工参与，并且隐马尔可夫模型中状态转移矩阵初始化比较困难。

基于隐马尔可夫模型方法最先是由Andrew McCallum等人于2000年提出的，在该方法中以待抽取信息的每一个属性作为马尔可夫模型中的一个状态，利用隐式马尔可夫模型进行信息抽取，它适用于结构化信息的抽取[19]；Soumya Ray等人提出了一种在信息抽取中以HMM表示句子结构的方法，并将该方法应用到生物医学领域[20]。在国内, 王胜等人提出了一种基于熵的马尔可夫模型算法，他们利用该算法抽取页面中的地址信息[21]；钟敏娟等人提出了一种基于多模板的隐马尔可夫模型信息抽取算法，它利用聚类的方法将训练集中的样本分类，以每个类为一个模板，然后利用马

尔可夫模型进行信息抽取[22]。

1.2.2 搜索引擎发展现状

第一代搜索引擎大约出现在 1994 年，Yahoo、AltaVista 是它们的代表。当时的互联网还没有这样庞大的信息，网络传输的速度也没有现在这么快，且网页类型主要以静态页面为主，页面中的数据类型也没有现在复杂，当时这类搜索引擎的主要目标是提高对整个 Web 的覆盖率，他们主要是以关键字搜索和目录式搜索的形式提供服务[23]，这类搜索引擎被定义为目录式搜索引擎（Directory Search Engine）。它主要是以人工或者半自动的方式搜集信息，由编辑员查看信息之后，人工形成信息摘要，并将信息置于事先确定的分类框架中。它的信息大多面向网站，提供目录浏览服务和直接检索服务，该类搜索引擎因为在处理过程中人工参与较多，所以信息准确、导航质量高，缺点是需要人工介入、信息量少、信息更新不及时[24]。

随着互联网的发展，互联网的信息量越来越大，传统的目录式搜索引擎已经不能很好的满足用户要求，其中最主要的原因是由于目录式搜索引擎需要人工参与，在处理信息速度方面远不能满足日益增长的数据量要求，这样对用户提交的查询请求，搜索引擎的虽然返回成千上万的查询结果，但是用户依然难以在查询结果中找出完整的需要的信息，从一定程度上影响了用户查询信息的完整性和准确性。为保证能够及时返回对用户真正有价值的网页，第二代搜索引擎引入了超级链接分析技术，并通过网络蜘蛛来自动采集 Web 上的信息，从而有效地加快信息处理速度，它们的主要代表是 Google。Google 提出了 PageRank 排序技术，其目标是尽可能的使搜索引擎用户能够在返回的查询结果最前面部分找到他们感兴趣的信息。

到目前为止，主流的搜索引擎多属于横向的水平型搜索，在现有的技术水平基础上，属于横向水平的搜索引擎在满足搜索信息量完整性的同时却难以兼顾查询的准确性，这样在查询结果中出现了大量与用户需要信息无关的查询结果，从而影响了搜索引擎的查找效率。新出现的第三代搜索引擎力求在自然语言处理、数据挖掘、机器自动学习技术、主题相关性等方面有所突破，而垂直搜索引擎正是第三代搜索引擎的代表。目前，有关垂直搜索引擎的研究正在成为一个热点研究领域，具有代表型的垂直搜索引擎系统有：

Cora Search Engine：它是由 Adrew McCallum 等人提出的一种面向计算机科学研究论文搜索的垂直搜索引擎，它利用 HMM 的信息抽取方法抽取计算机科学研究论文 Web 页面中的标题、作者和摘要等[25]。

Libra Academic Search：它是一种面向科研的垂直搜索引擎，在该搜索引擎中，它将研究者、科学论文、会议等都当成一种 Web 对象，从 ACM 数字数据库、DBLP 和 CiteSeer 等数据源抽取信息[26]。

Windows Live Product Search：它是一种从互联网抽取商品记录信息的垂直搜索引擎，主要应用的是 Libra Academic Search 中的技术[26]。

Libra Academic Search 和 Windows Live Product Search 都是微软公司的产品，其中 Windows Live Product Search 还处于测试阶段。他们在对 Web 页面中信息抽取中利用了 Peng Cai 等人提出的基于视觉分析的 VIPS 算法。

1.3 研究内容

本文主要研究如何 Web 页面中抽取出用户感兴趣的主题信息，并将它转化为结构化数据形式，图 1-1 描述了本文的主要工作。

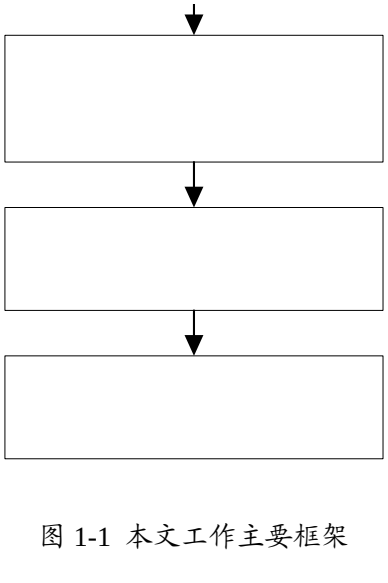

图 1-1 本文工作主要框架

具体的研究的内容如下：

1）针对目前 Web 信息抽取方法实现复杂等问题，本文对页面结构、主题属性页面显示格式与表示方式特征进行分析，设计了一种基于标签序列的 Web 页面主题信息抽取方法。该方法通过设定的策略，根据策略和样本建立了抽取规则，构建规则库，借助规则库实现对页面主题信息抽取。它降低了处理 Web 页面过程的复杂性，减少了页面处理时间。

2）目前分装器已经越来越多的应用到 Web 信息抽取中，当需要抽取的 Web 页面结构发生变化而出现主题的新属性时，采用目前方法建立的分装器并不能随着这种变化准确地发现这些新属性。针对该问题，本文根据待抽取属性与已抽取属性在页面中的特点，提出了一种基于可信度的 Web 页面主题新属性发现方法。该方法通过定义一系列规则和证据，量化待抽取属性需要抽取的可信度，再根据该可信度判断待抽取属性是否为主题的新属性。

3）设计了一种垂直搜索引擎原型系统，该原型系统主要特点是结合本文提出的基于标签序列的Web页面主题信息抽取方法和基于可信度的Web页面主题新属性发现方法，设计了一种专业网络蜘蛛，并采用 Lucene 开源软件将采集的信息建立索引，维护索引信息库，同时根据用户输入的条件，返回用户感兴趣的主题信息。

1.4 本文组织结构

本文共分为六章：

第一章，绪论。本章主要介绍了 Web 信息抽取和垂直搜索引擎的研究现状，以及本文的研究内容。

第二章，Web 信息抽取及垂直搜索引擎技术。本章介绍了本文在 Web 信息抽取将用到的部分技术，以及垂直搜索引擎技术的原理。

第三章，基于标签序列的 Web 页面主题信息抽取方法研究。本章主要针对目前 Web 信息抽取方法实现复杂等问题，设计了一种基于标签序列的 Web 页面主题信息抽取方法。

第四章，基于可信度的 Web 页面主题新属性发现。本章主要是针对目前分装器的不足，提出一种利用可信度发现 Web 页面主题新属性的方法。

第五章，垂直搜索引擎原型系统设计。本章主要介绍了该原型系统的设计。

第六章，总结和展望。对本文工作的总结，并指出将来需要更进一步努力的方向。

第二章 Web 信息抽取及垂直搜索引擎技术

Web 信息抽取技术成为了垂直搜索引擎的关键技术之一。本章主要介绍了本文需借鉴的 Web 信息抽取关键技术，在本章的最后还对 Lucene 全文搜索引擎开源软件和垂直搜索引擎技术的原理进行了介绍。

2.1 DOM 技术

2.1.1 DOM

根据 W3C 的定义，DOM(Document Object Model)[27]是一个允许程序和脚本动态地获取和更新文档内容、结构和风格的接口。1998 年，W3C 发布了 DOM Level 1，最新的版本是 2004 发布的 DOM Level 3。

DOM 规范包含两个关键的抽象：树状的层和用来表示文档文本以及结构的集合。它为有效的 HTML 和 Well-Formed XML 格式文件提供了应用程序的接口（API），并定义了用来加载 HTML 和 XML 文档的方法，它是一种独立于平台及语言的接口，允许对树结构的文档进行操作。一般在操纵文档之前必须在内存中生成 DOM 树，所以支持高性能的虚拟内存对于树型文档来说是非常必要的。DOM 定义了文档的逻辑结构以及存取和维护文档的方法，利用 DOM，程序员可以建立文档，遍历文档的结构，还可以增加、删除以及修改文档的元素和内容。

2.1.2 Cobra 软件

本文采用 Cobra[28]软件生成 DOM 树。Cobra 是开放源代码，遵从 LGPL 许可，纯 Java 的 HTML 映射与 DOM 解析器，支持对 HTML 4, Javascript 和 CSS 2 的解析；同时它还支持增量式的 HTML 映射，也就是一个 HTML 文档不需要全部加载它的源代码到内存中进行解析，当一个扩展的脚本在需要完全解析前加载的时候，这种机制非常有用。它最新的版本是 2008 年 3 月发布的 0.98.1。

Cobra 的使用比较简单，系统提供了一个 ParserTest 类说明如何在程序中使用 Cobra 对一个 HTML 文档建立 DOM 树。Cobra 中包含六个包，分别是：

1）org.lobobrowser.html 包：包含在解析过程中需要得到实施的背景界面，以便使用解析器和渲染器。

2）org.lobobrowser.html.domimpl 包：包含一系列基于 W3C HTML DOM Level 2 的实现接口。

3）org.lobobrowser.html.gui 包：提供一系列与 Java Swing 组件相兼容的接口，

以便于将显示 DOM 树。

4）org.lobobrowser.html.parser 包：这个包比较重要，它实现了解析 HTML 文档的类。

5）org.lobobrowser.html.renderer 包：包含了 HTML 渲染器的构造接口。

6）org.lobobrowser.html.test 包：包含 Cobra 前面几个包中类的测试软件类和简单的背景接口实现。

图 2-1 是利用 Cobra 对百度首页生成的 DOM 树：

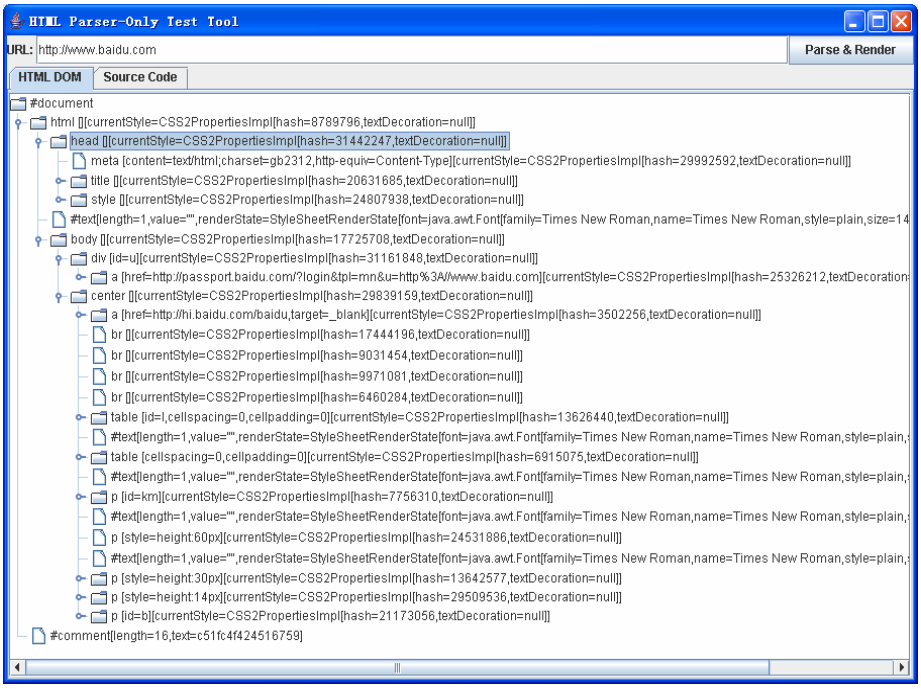

图 2-1 Baidu 首页的 DOM 树

2.2 分装器

目前分装器(Wrapper)[29]在信息抽取系统中应用的越来越多，一般的理解认为分装器是一个程序，它包含一系列规则，用于从特定的信息源中抽取相关内容，并以特定形式返回抽取结果。

对于不同的信息源，分装器的目的并不相同，在Web环境中，分装器的目的是以分装器中包含的规则将网页中符合规则的信息抽取出来，并以结构化的形式储存起来，以方便进一步的处理，它的衡量标准是抽取信息的准确性和完整性。

分装器的构建主要有人工、半自动和全自动三种方式，人工构造的分装器在信息的抽取方面有比较高的准确率，但是消耗人力资源比较多，利用效率比较低。目前分装器生成主要以半自动和全自动为主，典型的应用系统有W4F[30]、WIEN[31]、XWRAP[32]、Omini[33]、STAVIES[34]和MDR[35]。

W4F是使用一种HEL(HTML extraction language)查询语言去发现页面中待抽取的信息，利用HEL查询语言，人工标记页面中需要抽取的信息，从而构造分装器中的规则，并将抽取出的信息以XML文档的形式表现出来；WIEN系统使用了自主学习的方法，从一定程度上降低了人工的参与，WIEN系统的处理过程分为五个步骤，分别是主题的指定、收集和标注样本、分装器构造、信息源模式选择和信息的抽取，其中主题的指定、收集和标注样本这两个过程需要人工指导，以标记页面中需要抽取信息的属性，信息源模式选择是选择待抽取的信息源，在Web环境中为HTML文档；XWRAP提供了一个与人交互的图形化界面，它将以XML标记语言的形式生成分装器中的规则，它的另外一个贡献是能对网页源代码是进行语法检测，并能修正其中的错误语法。Omini系统将抽取过程分为两个阶段，首先它将HTML文档解析成一棵DOM树的形式，并利用一系列子树抽取算法定位DOM树中最小的子树，该系统认为感兴趣的信息就隐藏在子树中；其次它使用一组实体抽取算法去发现合适的实体分隔标记，这些标记将待抽取的信息分离出来；MDR对页面建立网页结构树，通过对树中的临近的节点进行拼接，通过节点序列对比将文档分成不同的相似区域，它认为文档中的主题信息在这些相似区域中；STAVIES主要利用文档的格式特征，使用聚类方法发现文档中用户感兴趣的信息。

以上方法在抽取页面主题信息时，需要一定的先验知识作为指导，如MDR在抽取主题信息时，需要认为它发现的区域包含特殊的字符，这个字符确定需要人工的参与。

本文提出了一种基于标签序列的Web页面主题信息抽取方法，它通过对页面结构、主题属性页面显示格式和表示方式特征进行分析，并提出一种策略，构建分装器的规则，实现Web页面主题信息的抽取。

2.3 Web 信息抽取评价指标

由于 Web 信息抽取属于信息抽取的一个分支，本文采用的是 MUC[36](Message Understanding Conference)会议定义的评价指标。MUC 由美国国防高级研究计划委员会(DARPA, the Defense Advanced Research Projects Agency)资助，它的显著贡献在于定义了信息抽取系统的评价指标。

在MUC中，衡量信息抽取系统的性能主要根据两个评价指标：召回率(Recall)和准确率(Precision)[37]，召回率等于抽取正确记录数占应有记录数的比例；准确率等于正确抽取记录数占抽取所有记录数的比例，具体定义如下：

$$召回率=\frac{抽取正确记录数}{应有记录数} \qquad 公式（2\text{-}1）$$

$$准确度=\frac{正确抽取记录数}{抽取所有记录数} \qquad 公式（2\text{-}2）$$

通常对于一个信息抽取系统来说，召回率和准确率不可能两全其美，当召回率高时，准确率低，准确率高时，召回率则比较低；为综合评价系统性能时，有时需要同时考虑召回率和准确度，MUC提出F指数来衡量系统的性能，其定义如下：

$$F=\frac{((beta)^2+1.0)*P*R}{((beta)^2*P)+R} \qquad 公式（2\text{-}3）$$

其中 P 指准确率，R 指召回率，beta 是召回率 R 和准确率 P 的相对权重，beta 等于 1 时，二者同样重要；beta 大于 1 时，准确率更重要一些；beta 小于 1 时，召回率更重要一些。在 MUC 系列会议的定义中，beta 取值一般为 1、1/2、2。

2.4 文本预处理

2.4.1 文本分类

文本分类是自然语言理解和人工智能一个重要的领域，它是文本挖掘的重要组成部分，对提高信息检索的速度和准确率方法十分重要，目前已经有了很多的文本分类方法，如贝叶斯分类算法[38]、决策树分类算法[39]、支持向量机[40]和神经网络分类[41]方法，一般的文本分类处理流程如图 2-2 所示，文本分类大致可以分为三个过程，分别是文本表示过程, 训练样本学习过程和文本类别判断过程。

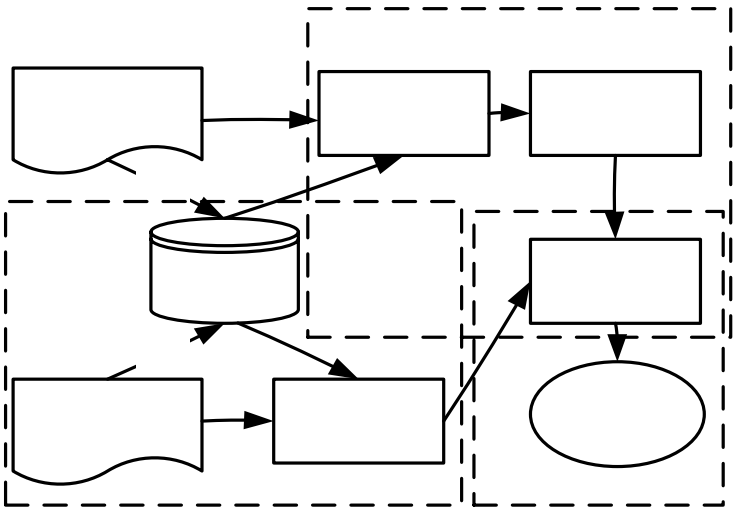

图 2-2 文本分类一般处理流程

1）文本表示过程

文本表示过程是为了方便文本被计算机处理，对文本采取的特定表示方式，目前向量空间模型 VSM[40]（Vector Space Model）是近年来应用较多且效果较好的文本表示方式之一。在该模型中，文档被看做是一组正交词条所生成的向量空间，每个文档 d 可以由向量 V(d)表示，其描述形式如下：

$V(d)=((t_1, w_1(d),\ldots\ldots,(t_i,w_i(d)),\ldots\ldots(t_n,w_n(d))$

其中 t_i 为词条项，t_i 的选择是一个特征选择的过程，特征选择是将文本中对表达文本所属类别有比较强说服力的词汇从文本中抽取出来，形成一个向量，在特征选择的过程中还需要去掉对表达文章类别不太重要的词汇，如中文文本中的“的”、“地”、“得”、“着”、“了”等停用词。

$w_i(d)$表示词条项 t_i 在 d 中的权重，$w_i(d)$一般定义为 t_i 在 d 中出现的频率 $tf_i(d)$ 的函数，即 $w_i(d)=f(tf_i(d))$，一个典型的计算词条项权重的公式如下：

$$w_i(d)=\frac{tf_i(d)\log(\frac{N}{n_k}+0.01)}{\sqrt{\sum_{k=1}^{n} tf_i(d)^2 * \log^2(\frac{N}{n_k}+0.01)}} \qquad \text{公式（2-4）}$$

其中 N 表示训练集中样本总数，n_k 为词条项 t_k 的文本频数。

2）训练样本学习过程

训练样本学习过程是对训练集中的样本进行分析，构造分类器，即将训练集中的样本表示为向量 V(d)的集合，集合中每一个向量代表一个分类。

3）文本类别判断过程

文本类别判断过程是根据训练样本构造的分类器，采用一定的策略，判断文本所属的类别。

2.4.2 中文分词

中文分词[42]是东亚等语种特有的一个处理过程，它是信息抽取、信息检索、机器翻译、文本分类、自动文摘、语音识别和文本语音转化等中文信息处理领域的基础研究课题。

目前的分词方法主要是基于词典的分词方法，它有三个要素，分别是：分词词典、文本扫描顺序和匹配原则[43]。

分词词典是一个语料库，它包含对各个单词和短语的词性标注。

文本扫描顺序是对需要分词的文本进行初步处理，以判断文本怎样进行分词，它正向扫描、逆向扫描和双向扫描三种方式，其中正向扫描是指从待切分文本的开头开始扫描，逆向扫描是指从待切分文本的末尾开始扫描，双向扫描是正

向扫描和逆向扫描的结合，即同时对待切分文本进行正向扫描和逆向扫描。

匹配原则是中文分词准确性的关键，目前已经有很多匹配的方法，主要有最大匹配、最小匹配、逐词匹配和最佳匹配方法。最大匹配法的基本思想设定一个最大的词的长度，对文本中词组以最大的长度进行逐字进行分词，最终找到文本中所有的词；最小匹配法的基本思想是使待切分语句分词后得到的词最少；逐词匹配法是指把词典中的词按由长到短的顺序在待切分语句中进行搜索和匹配，直到把所有的词都切分出来为止；最佳匹配法的基本思想是词典中的词条按照词频的大小顺序排列，以求缩短分词词典的检索时间，从而降低分词的时间复杂度，它也是一种基于统计的分词方法。

2.5 Lucene 工具包

Lucene[44]是一个全文搜索引擎软件工具包，但它并不是一个全文搜索引擎，自 2000 年 Doug Cutting 在 SourceForge 上公布源码以来，它获得了迅速的发展，目前 Lucene 属于 apache 软件基金会 jakarta 项目组中的一个子项目，最新的 java 版本是 Lucene Java 2.3.0。

Lucene 的主要功能是提供一组通用的 API，以便于其他系统使用它添加信息索引和搜索功能，Lucene 提供的接口生成的数据不同于数据库中的表，它是以文件的形式将索引信息存于磁盘中，这样基于 Lucene 的搜索引擎系统的索引信息库就与具体的数据库系统无关。同时 Lucene 提供了多种查询方式，如在查询中可以使用布尔运算等。

Lucene 源码中主要包含七个包，核心包有三个，分别是 org.apache.Lucene.index 包、org.apache.lucene.search 包和 org.apache.lucene.analysis 包。其中 org.apache.lucene.index 包的主要功能是管理采集信息的索引，包括索引的建立和维护；org.apache.lucene.search 包的功能是检索索引库中的信息，同时还包括分析用户的查询条件；org.apache.lucene.analysis 包中主要是进行一些语言分析，如对于中文的，提供中文分词功能。

2.6 垂直搜索引擎原理

垂直搜索引擎是通用搜索引擎的细化和延伸，二者基本工作原理基本相同[24]，图 2-3 描述了垂直搜索引擎的工作原理[24]。

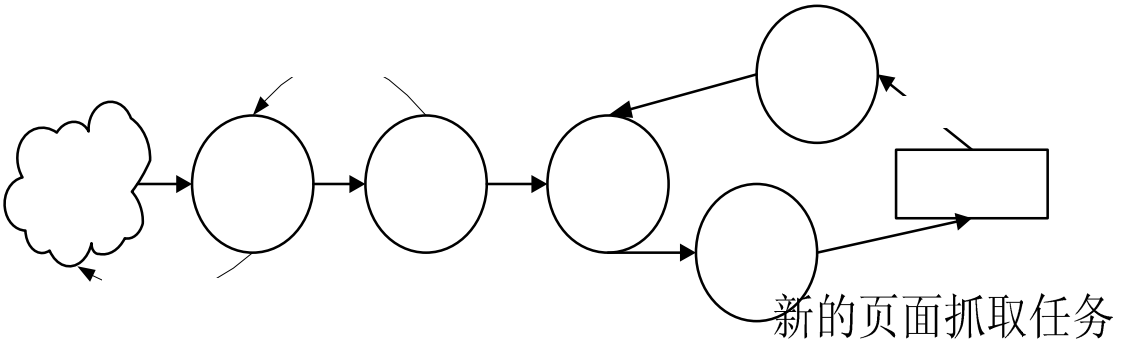

图 2-3 垂直搜索引擎的工作原理

从图 2-3 可以看出，垂直搜索引擎主要包含专业网络蜘蛛、页面信息处理和信息索引与检索三个典型模块。

1）专业网络蜘蛛

网络蜘蛛的功能[24][25][45]是从 Web 上采集信息。搜索引擎的性质和网络蜘蛛的采集信息的策略紧密联系在一起。为了获得比较高的 Web 覆盖率，通用搜索引擎的网络蜘蛛并不关注采集的信息是否满足用户的要求，它试图对 Web 中整个拓扑图的每个节点都遍历到[46]。网络蜘蛛遍历方式一般有三种，即深度优先、广度优先和两种遍历方式混合的遍历方式，专业网络蜘蛛的遍历方式也通常是这三种。

垂直搜索引擎由于是服务于特定人群，它对采集的页面内容具有选择性，仅限于特定主题，因此在采集信息的过程中没有对整个 Web 进行遍历的必要，只需选择与特定主题相关的页面进行采集。它是通用网络蜘蛛的特殊运用，两者的基本原理是相同的，但是在处理的过程中采用的策略，专业网络蜘蛛和通用网络蜘蛛是不同的，同时专业网络蜘蛛在搜索 Web 时，需要对已发现的链接进行主题预测和识别，对网页是否与特定主题相关做判断，以决定对该链接的处理方式。其基本原理图如 2-4 所示：

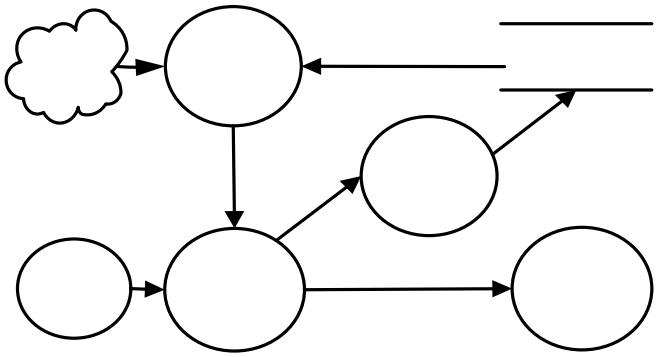

图 2-4 专业网络蜘蛛原理图

2）页面信息处理

页面处理模块对专业网络蜘蛛提取的网页信息进行处理，首先它需要对采集

的信息进行过滤，去掉页面中的广告、导航链接等页面噪声信息。其次需要对页面中提取出页面中的关键字，以便于判断页面是否属于特定主题，并将提取出的关键字进行分类等处理，以存入索引信息库，方便用户的查询。

在页面信息处理模块中，如果采集的是英语等西方语种的页面时，不需要进行分词处理，但是对于中文等东亚文字而言，还需要对采集的页面信息进行分词处理，以方便对采集的信息进行分类和建立索引。

3）信息索引与检索模块

信息索引与检索模块包含两个子模块，分别是信息索引子模块和检索子模块，信息索引与检索模块原理图如图 2-5 所示。信息索引子模块对采集的页面信息建立索引，存入索引信息库，并随着新信息的入库，对原索引信息库中的信息进行维护，并根据用户输入的查询关键字，在索引库中快速检出文档，进行文档与查询的相关度评价，对将要输出的结果进行排序，并将查询结果返回给用户；检索子模块是用户检索过程，它对前两个过程的检验，首先分析用户的查询请求，得到用户感兴趣的条件区域，将它提交给信息索引子模块，根据信息索引子模块返回的结果，以特定的显示风格呈现给用户。

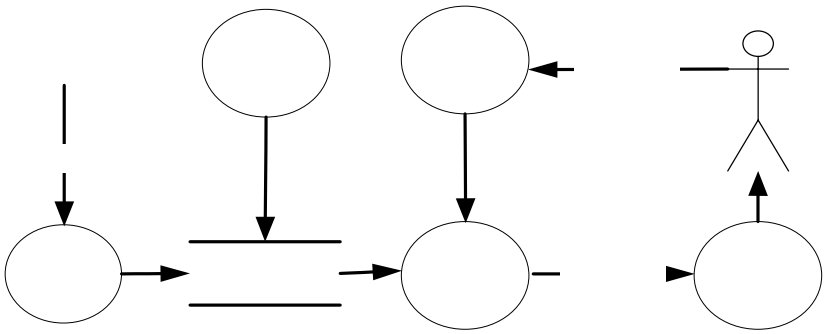

图 2-5 信息索引与检索模块原理图

2.7 本章小结

本章对将采用的 Web 信息抽取的关键技术进行了介绍，具体有 DOM、分装器、文本预处理以及 Web 信息抽取技术采用的衡量标准；由于本文实验用到的大部分是中文网页，本章还介绍了中文分词；最后本章还介绍了 Lucene 开源软件和垂直搜索引擎技术的原理。

索引信息库维护

第三章 基于标签序列的 Web 页面主题信息抽取方法研究

3.1 引言

现今的 Web 信息抽取方法，主要有基于 HTML 结构的信息抽取方法[9-14][47-52]、基于 Ontology 的信息抽取方法[15-18][53-55]、基于隐马尔科夫模型的信息抽取方法[19-22]，基于自然语言处理的信息抽取方法[5-7][57-59]。

不同的抽取方法实现方式不同，从而对于不同的 Web 页面有着不同的抽取效果。基于 HTML 结构的信息抽取方法通常对 Web 页面建立网页结构树，然后对该树的节点进行分析，通过聚类、相似性判断或者对树的节点进行语义性标注，实现 Web 页面的信息抽取，目前基于此方法的 Web 信息抽取系统大多通过一系列的启发式算法实现了一定程度的自动化，但是利用启发式算法抽取信息一般需要比较长的页面处理时间，同时该方法通常只能判断 Web 主题信息所在区域，不能准确对区域中的主题信息和噪声等进行区分，这样从一定程度上影响了信息抽取的准确性。基于 Ontology 的信息抽取方法和基于自然语言处理的信息抽取方法主要面临两个问题，一是网页信息显示格式的清洗，二是知识库的建立，对于第一个问题，网页中信息显示的格式千差万别，目前暂时还没有出现在缺乏人工指导的前提下有效分离出有用信息的方法；对于第二个问题，知识库的建立一般需要专业人士参与，并且由于 Web 更新速度快，知识库经常不能及时更新，所以该方法虽然能比较准确的抽取出 Web 页面的信息，但是抽取 Web 页面信息开销却比较大。基于隐马尔科夫模型的信息抽取方法需要较多的人工参与，并且由于在对样本训练过程中没有有效的先验知识作为指导，隐马尔可夫模型中状态转移矩阵初始化比较困难，并且也需要对网页信息显示格式进行清洗。

从上面分析可以看出，目前的 Web 信息抽取方法虽然取得了很多的成果，但是仍存在以下问题：

1、准确度高的信息抽取方法大多需要人工参与，并且需要建立复杂的知识库等，其实现过程比较复杂。

2、自动化程度高的信息抽取方法一般页面处理时间比较长，并且准确度比较低。

针对以上问题，本章对页面结构、主题属性页面显示格式与表示方式特征进行分析，设计了一种基于标签序列的 Web 页面主题信息抽取方法。该方法通过设定的策略，根据该策略不需要建立复杂的知识库等，从而降低了处理 Web 页面过程的复杂性。并根据策略和样本建立抽取规则，构建规则库，借助规则库实

现对页面主题信息抽取，由于根据规则抽取页面主题信息过程中不包含启发式算法，从而减少了页面处理时间。并通过将此方法应用在部分网站的页面中进行实验，获得了比较好的效果，具有较好的实用性。

本章 3.3 节中建立了该方法相应的模型，并通过该模型具体说明了该方法的实现方式。

3.2 相关概念与分析

3.2.1 相关概念

为了更好的说明本文的方法，我们首先对本文将要用到的主要概念进行阐述，其具体定义如下：

定义 3-1 主题：它表示为人们感兴趣的客观事物，如手机参数。

定义 3-2 属性名：它表示主题某个特征的具体名称。

定义 3-3 属性值：它表示主题某个特征的具体内容，如属性手机名它的属性值可以是诺基亚 1208 等具体手机名称。

定义 3-4 主题属性：它表示主题的某个特征，由属性名和属性值组成，如手机参数主题属性：（手机名, 诺基亚 1200）。

定义 3-5 主题信息：它表示某一特定主题属性值的集合，如一条典型的手机参数主题信息可以为：(诺基亚 1200,200,2007, GSM 900/1800, 直板，单色屏,68×96 像素, 不支持, 无摄像头)。

定义 3-6 噪声：它表示页面中与主题无关的信息。

定义 3-7 主题属性记录：它是某一主题属性名的集合，如手机参数主题属性记录（手机名，参考价格，上市时间，手机制式，手机外形，主屏参数，数据业务，摄像头）。本文认为对某一主题，其主题属性记录唯一，多个主题属性记录构成主题属性配置文件。

定义 3-8 标签：它是 HTML 语言的基本单位，通常用两个角括号环绕，它有三种描述形式如下：

tag=<tagName>或 tag=</tagName>或 tag=<tagName/>

其中 tagName 标签名，<tagName>形式表示开始标签，如<html>，</tagName>形式表示结束标签，如</html>，<tagName/>表示独立标签，如
。

3.2.2 页面结构分析

目前，网页中包含各种各样的信息，一般来说，各个页面都含有页面主体内容、导航信息、友情链接以及广告等四个部分，而通常为了更好的使网页中的信

息呈现给网页浏览用户，网页提供者都会对页面中出现的信息进行简单的分类，使得各种信息在页面中具有视觉上的连续性，如图 3-1 所示的两个页面表示了公司联系方式和手机参数两类主题信息，它们都在连续的区域出现，根据这个特点，研究人员提出了一系列 Web 信息抽取的方法，文献[14]所提出的基于视觉分析的 Web 信息抽取方法就是对这种网页结构特点典型应用，本文将这种特点称为主题信息视觉区域连续性。

图 3-1 两类主题信息视觉上所在区域

网页中文本等元素能在浏览器中显示出来，是由于定义一系列表达这些的页面元素的 HTML 标签，这些标签与网页中需要显示的文本等元素构成页面的源代码。为有效地抽取网页中信息，我们需要对网页源代码进行分析。在网页源代码中，主题信息都被标签环绕，图 3-2 表示图 3-1 中网页对应的源代码，从图 3-2 我们可以看出，主题信息中的各个属性之间除了相隔有限个 HTML 标签外，并没有其他的信息，它们出现在页面源代码一个紧密联系的区域中，本文将这种特点称为主题信息源代码区域连续性。

```
<ul class="mainTextColor">
    <li>电    话：86 0732 8270798</li>
    <li>移动电话： 13507327328</li>
    <li>传    真：86 0732 8270798</li>
    <li>地    址：中国 湖南 湘潭市 韶山东路 46 号 2 号门</li>
    <li>邮    编：411100</li>
    <li>公司主页：<a class="topicLink draft_no_link" href="http://www.sanlijc.com.cn" target="_blank">http://www.sanlijc.com.cn</a><br/><a style="margin-left:67px" class="topicLink draft_no_link" href="http://sanlijichuang.cn.alibaba.com" target="_blank">http://sanlijichuang.cn.alibaba.com</a></li></ul>
```

```
<dl>
  <dt>诺基亚 1200</dt>
  <dd><strong>参考价格：</strong>
      <spanclass="red"
      id="ppckbj">-</span></dd>
  <dd><strong>上市时间：</strong>2007
  </dd>
  <dd><strong>网络制式：</strong>GSM
  900/1800 </dd>
  <dd><strong>手机外形：</strong>直板
</dd>
  <dd><strong>主屏参数：</strong>单色屏,68×96 像素</dd>
  <dd><strong>数据业务：</strong>不支持
  </dd>
  <dd><strong>摄像头：</strong>无摄像头
  </dd>
</dl>
```

图 3-2 两类主题信息 HTML 源代码中所在区域

根据以上两个特点，我们在抽取页面中主题信息时，就只需要关注页面源代码中某一段连续的源代码，通过对这段源代码进行分析，分离出标签名之间的文本片段，从而抽取出页面中主题信息。本文设计的基于标签序列的 Web 页面主题信息抽取方法正是根据这两个特点实现 Web 页面中主题信息的抽取，它通过设定的策略将主题信息所在区域的源代码中标签名抽取出来，形成主题区域标签序列；同时对待抽取的整个页面也根据这种策略生成页面标签序列，通过判断页面标签序列中是否包含主题区域标签序列，以决定是否抽取 Web 页面主题信息。

3.2.3 主题属性页面显示格式特征分析

为了方便用户获取页面中的主题信息，网页提供者通常对显示在页面中的主题信息进行组织，以一定的格式使得主题信息在页面中表现出来。通过对部分网站的页面进行分析，我们发现大部分的主题信息中属性名和属性值在页面源代码中的表现形式有如下四种：

1、<tag>主题属性名+[属性名格式特征]</tag>……{<tag>属性值</tag>}

其中 tag 表示标签名，[属性名格式特征]为可选项，{<tag>属性值</tag>}表示页面源代码中可能含有 0 个或多个这样的选项。如在图 3-1 中第二个页面中参考价格属性名及属性值的在页面源代码中表现形式为：参考价格：-，其中参考价格属性名格式特征为属性名后带有“：”特殊符号，其属性值为“-”。

2、<tag>主题属性名+[属性名格式特征]+属性值</tag>

其中 tag、[属性名格式特征]与第一种形式含义一样。如在图 3-1 中第一个页面中电话属性名及属性值的在页面源代码中表现形式为： 电   话：86 0732 8270798，其中电话属性名格式特征为属性名之间含有特殊符号” ”和“：”，其属性值为 86 0732 8270798，并且传真、地址和邮编的属性名格式特征与电话的相同。

3、<tag><tag>主题属性名+[属性名格式特征] </tag>属性值</tag>

其中 tag、[属性名格式特征]与第一种形式含义一样，在图 3-1 中第二个页面中上市时间属性名及属性值的在页面源代码中表现形式为：<dd>上市时间：2007</dd>，其中上市时间属性的格式特征为属性名后带有“：”特殊符号，其属性值为 2007。

4、<tag>属性值</ tag>

其中 tag 与第一种形式含义一样，如图 3-1 中第一个页面中手机名属性，在页面源代码中表现形式为：<dt>诺基亚 1200</dt>。

3.2.4 主题属性页面表示方式特征分析

对同一属性在不同的页面有着不同的表示方式，如图 3-3 中第一个页面中手机制式属性在第二个页面中显示为网络制式，为了解决这个问题，本文通过一个映射函数，将页面中显示的属性名映射到标准属性名，其描述形式如下：

$$f(attrN_i) = STANDARD\_ATTR_j$$　　公式（3-1）

本文中我们称 $attrN_i$ 为伪属性名，该属性名不含页面显示的属性格式特征；$STANDARD\_ATTR_j$ 为标准属性名，它表示主题属性记录中的属性名。

图 3-3 同一属性不同页面中属性名的表示方式

属性名在页面源代码中为一个字符串，但是这个字符串中可能含有特殊符号，如 3-3 第二页面中，参考价格属性在页面源代码中可以由字符串“参考价格：”表示，在这个字符串中，除了包含属性名本身的名称外，还含有特殊字符“：”，为了和标准属性名对应，必须去掉特殊字符“：”，本文将属性名在页面与源代码中显示的方式定义为页面属性格式记录，其描述形式如下：

attr=(attrName,attrFeature)

attrName 表示页面显示的属性名，它包含属性名格式特征，attrFeature 为属性名格式特征，将页面中显示的属性名去掉属性名格式特征的过程称为属性清洗过程，页面中显示的属性名通过此过程后形成伪属性名。

3.3 一种基于标签序列的 Web 页面主题信息抽取方法

3.3.1 相关策略

页面源代码中的标签规定了浏览器显示文档元素的格式，大多数标签要求必须成对出现，它们分别为开始标签和结束标签，而有些标签则没有做此要求，再加上当前技术人员编程的一些不规范性行为，使有些应当有结束标签的标签没有结束标签，这对网页的处理工作带来了很大的不便；同时页面源代码中所有的注

释和 script 标签不对页面中其他元素显示造成任何影响。

根据以上观点，本章方法拟采用如下策略：

策略 1：忽略所有标签的结束标签。

策略 2：标签 L_i 的内容 C_i 定义为前一个标签 L_i 和后一个标签 L_{i+1} 之间的文本。

与文献[14][56]类似，本文根据策略 1、2，定义页面的标签和它的内容为一个基本对象。

策略 3：忽略 HTML 文档源代码中所有的注释和 script 标签。

3.3.2 相关定义

本节主要对本章拟用到的专业标识语进行了定义，如下：

定义 3-9 主题属性格式文件：它是页面属性格式记录的集合，其描述形式如下：

$F=\{attr_1,attr_2,\ldots\ldots,attr_i,\ldots\ldots,attr_n\}$

其中 $attr_i$ 表示主题属性格式文件中第 i 个页面属性格式记录。

定义 3-10 标签序列：它表示一类页面的结构，其描述形式如下：

tags=tag0*tag1*……*tagi*……*tagn

其中 tagi 表示页面中的第 i 个标签名。如对于页面<html><title></title><body>WelcomeHello<body></html>其对应的页面标签序列为：html*title*body* font*b。

定义 3-11 主题区域标签序列：它是页面中的一段标签序列，该段标签序列中标签的内容，包含页面需要抽取的主题信息，其描述形式如下所示：

area=tagi.....*tagj*……tagk

其中 tagi 表示该区域的第一个标签，tagk 表示该区域的最后一个标签，如图 3-1 中第二个页面的主题区域标签序列为：dt*dd*strong*span*dd*strong*dd*strong*dd*strong*dd*strong*dd*strong*dd*strong。

定义 3-12 主题区域标签位置：它表示标签在主题区域标签序列中的位置，并判断该位置标签的内容是否包含主题属性，其描述形式如下：

p = (i, attrType, contentType, isNoise)

其中 i 表示主题区域标签序列中第 i 个标签，attrType>0 表示该标签内容包含主题属性名，attrType 的值对应主题属性格式文件中第 attrType 个页面属性格式记录，当 attrType=0 时，表示该标签内容不包含主题属性名；contentType 与含义与 attrType 类似，当 contentType>0 表示该标签内容包含主题属性的属性值，contentType 的值对应主题属性格式文件中第 contentType 个页面属性格式记录，

contentType=0 表示该标签内容不包含属性值；isNoise=1 表示该标签内容为噪声，isNoise=0 时表示该标签内容不是噪声。如定义 3-10 例子中主题区域标签序列第一个主题区域标签位置为：（1，0，1，0），它表示该标签的内容只包含主题属性格式文件中第一个页面属性格式记录表示属性的属性值。

定义 3-13 位置向量：它是主题区域标签位置的集合，其描述形式如下：

$pos=\{p_1\ldots\ldots,p_i,\ldots\ldots,p_n\}$

其中 p_i 表示第 i 个主题区域标签位置，pos={(-1,-1,-1,-1)}表示页面没有需要抽取的信息。

定义 3-14 规则：它是一个四元组，其描述形式如下：

r=(area,pos, F,source)

其中 area 表示主题区域标签序列，pos 表示位置向量，F 表示主题属性格式文件，source 表示样本所在网站的域名。

定义 3-15 规则库：它是规则的集合，其描述形式如下：

$R=\{r_1,r_2,\ldots\ldots,r_i,\ldots\ldots,r_n\}$

其中 r_i 表示规则库中第 i 条规则。

定义 3-16 内容向量：它是页面标签内容的集合，其描述形式如下：

$Content=\{c_1,c_2,\ldots\ldots,c_i,\ldots\ldots,c_n\}$

其中 c_i 表示页面中第 i 个标签的内容。如在定义 3-10 中的页面的内容向量是 Content={“”,“”, “”,“Welcome”,“Hello”}。

定义 3-17 待抽取页面：它是一个三元组，表示一个将要抽取的页面，其描述形式如下：

W=(tags, Content, url)

其中 tags 是根据待抽取页面的源代码生成的标签序列，Content 是待抽取页面的内容向量，url 表示该页面的链接地址。

定义 3-18 待抽取区域：它是一个三元组，表示待抽取页面中一个可能包含主题信息的区域，其描述形式如下：

A=(tags,start,end)

其中 tags 为页面标签序列的一个子串，该子串中标签的内容可能包含主题信息，start 表示 tags 中第一个标签在页面中的位置，end 表示 tags 中最后一个标签在页面中的位置。

定义 3-19 主题信息向量：它是一个二元组，其描述形式如下：

$Info=((contentType_1,content_1),\ldots\ldots,(contentType_i,content_i),\ldots\ldots,(contentType_n,content_n))$

其中 $contentType_i$ 表示主题属性格式文件中第 $contentType_i$ 个页面属性记

录，$content_i$ 表示该记录对应属性的属性值。

3.3.3 基于标签序列的 Web 页面主题信息抽取模型

根据 3.2 节中的分析以及 3.3.1 节提出的策略，本章基于标签序列的 Web 页面主题信息抽取方法建立的模型如图 3-4 所示。

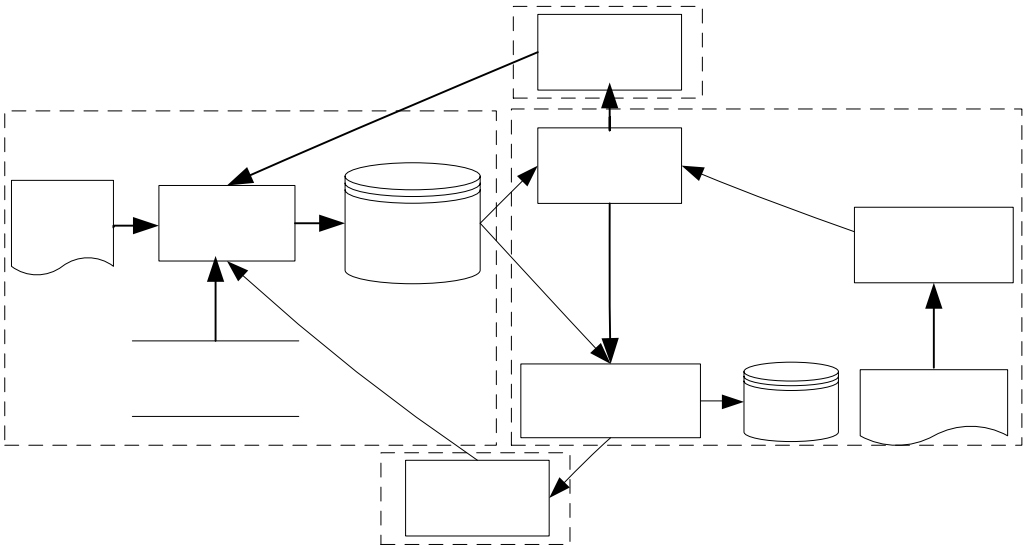

图 3-4 基于标签序列的 Web 页面主题信息抽取模型

该模型主要包含三个模块，分别是样本训练模块、主题信息抽取模块和新属性发现模块。

样本训练模块是通过对训练集中的样本进行分析，根据 3.3.2 节中的定义生成每个样本的主题区域标签序列、位置向量以及主题属性格式文件，并根据样本链接地址生成该样本所在网站的域名，从而归纳出页面抽取规则，并将规则存入规则库中。

主题信息抽取模块首先通过本文 3.3.1 节中设定的策略生成页面标签序列，然后将页面标签序列和规则中主题区域标签序列进行比较，生成待抽取区域集合，再对待抽取区域集合中每个待抽取区域包含标签的内容进行分析，通过规则中主题属性格式文件和位置向量判断该区域标签的内容是否包含主题信息，以判断该区域标签的内容是否需要抽取，从而实现页面中主题信息的抽取，并将抽取出的信息存入主题信息库。

新属性发现模块是进行主题信息抽取时，若没有发现页面中的主题信息，此时，很可能是页面中出现了新的主题属性。我们通过本文第四章中方法发现页面中是否有主题的新属性，当有新属性时，通过样本训练过程，归纳新的页面抽取规则，从而实现新结构页面主题信息的抽取。

下面对分别对该模型的样本训练模块和主题抽取模块进行具体介绍。

3.3.4 样本训练

样本训练模块的目的是生成规则库中规则。目前有很多的样本训练方法，大致可以分为以下三种，分别是机器学习的方法、自然语言理解的方法和人工指导的方法。机器学习的方法是对大量样本进行对比分析，进而归纳出规则，但是页面中的主题信息随机的出现在页面的区域中，如果没有人工指导，机器很难发现主题信息在页面中的位置。二是通过自然语言理解的方法归纳规则，但是该方法通常需要标注巨大的知识库，而知识库的建立一般需要领域专家的参与。为了获得比较好的效果，以上两种方法大多需要人工参与，基于这个原因，本文采用人工指导的方法对训练集的样本进行分析，生成页面抽取规则，其具体步骤如下：

1、 应用本文 3.3.1 节建立的策略，建立页面标签序列和内容向量，其过程如图 3-5 算法所示。并根据样本链接地址得到该样本所在网站的域名。
2、 通过人工指导，找出页面中主题信息所在区域，生成主题区域标签序列，如图 3-1 第二个页面的主题区域标签序列为：dt*dd*strong*span*dd*strong*dd*strong*dd*strong*dd*strong*dd*strong。
3、 根据内容向量判断主题信息所在区域标签的内容包含的是属性、属性值还是噪声，根据本文 3-14 中定义生成主题区域标签位置，并生成页面属性格式记录。
4、 将第三步生成的主题区域标签位置组成位置向量，将生成的页面属性格式记录存入主题属性格式文件。并结合第二步生成的主题区域标签序列和第一步生成的网站域名，构建抽取规则，并将规则存入规则库。

```
输入：
网页源代码：sourceCode
输出：
页面标签序列：tags
内容向量集合：Contents
GenTagsAndContents(sourceCode ){
1、 tags=“”,done=true,position=0;//position 代表每个标签的位置
2、 While(true)
      //获取 sourceCode 中第一个标签名 tagName
     tagName=sourceCode.FirstTagName();
     IF tagName==null break;//如果没有取到 tagName，则终止循环
     //如果 tagName 为 script 标签或者<!—标签
     IF tagName.LikePattern(“script”)||tagName.LikePattern(“<!--”)
```

```
        //获取 sourceCode 中第一个标签名 tagName 的位置 startIndex
        startIndex=sourceCode.FirstIndexOf(tagName);
        //获取 sourceCode 中下一个标签名 tagName
        tagName= sourceCode.NextTagName();
        //获取 sourceCode 中下一个标签名 tag 的位置 endIndex
        endtIndex=sourceCode. NextIndexOf (tagName);
        //将 sourceCode 中 startIndex 和 endtIndex 中间的子串替换为空串
        sourceCode=sourceCode.subString(0,startIndex)
                        +sourceCode.subString(endIndex);
        continue;
    EndIF
    //如果 tagName 的形式如</tagName>
    IF tagName.LikePattern("</tagName>")
      startIndex=sourceCode.FirstIndexOf(tagName);
      //将 sourceCode 中</tagName>的子串替换为空串
       sourceCode=sourceCode.subString(0,startIndex)
                    +sourceCode.subString(startIndex+"</tagName>".length);
       continue;
    EndIF
3、EndWhile
4、While(done)//循环生成标签序列
     tag=sourceCode.FirstTagName()
    IF tag==null   break;//如果没有取到标签名，循环终止
    ElSE   //如果取得标签名 tag
       tags=tags+ tag+"*";
   EndELSE
  startIndex=sourceCode.FirstIndexOf(tag);
  endIndex= sourceCode.NextIndexOf(tag);
  //如果找到下一个标签，则取出两个标签间的文本 txt
  IF endIndex>0
       txt= sourceCode.subString(startIndex,endIndex);
   EndIF
    ELSE//如果没有找到下一个标签，则将第一个标签后所有的文本 txt
        //存入 Contents 向量中
```

```
        txt= sourceCode.subString(startIndex);
        done=false;
    EndELSE
    //将 txt 存入 Concents 向量中
    Contents[position]=txt;
    //根据 endIndex 将 sourceCode 串分成两个子串，将 sourceCode 替换
    //为 endIndex 位置后的子串
    IF EndIndex>0
        sourceCode=sourceCode.subString(endIndex);
    EndIF
    position++;
  EndWhile
5、return tags，Contents。
}
```

图 3-5 标签序列和内容向量生成算法

在样本训练模块中第三步主要有两个问题，一是页面属性格式记录的生成，二是主题区域标签位置的生成。对第一个问题，在前面我们说明了将标签和其内容定义为一个基本对象，页面属性格式记录由标签内容决定，并通过分析，抽取标签内容中的属性名、属性名格式特征；对第二个问题，由于我们通过第二步已经得到了主题区域标签序列，接着是判断每个标签在标签序列中的位置，每个标签的内容包含的是属性、属性值还是噪声。如图 3-1 中第二个页面所标记区域的手机参数主题包含的属性有（手机名，参考价格，上市时间，网络制式，手机外形，主屏参数，数据业务，摄像头），则页面属性格式记录和对应属性的主题区域标签位置如下：

手机名属性格式记录：（“”，“”）

手机名属性所在标签位置：（1，0，1，0）

参考价格属性格式记录：（“参考价格”，“:”）

参考价格属性所在标签位置：（3，2，0，0）

上市时间属性格式记录：（“上市时间”，“:”）

上市时间属性所在标签位置：（6，3，3，0）

网络制式属性格式记录：（“网络制式”，“:”）

网络制式属性所在标签位置：（8，4，4，0）

手机外形格式记录：（“手机外形”，“:”）

手机外形属性所在标签位置：（10，5，5，0）

主屏参数属性格式记录：（“主屏参数”，“：”）
主屏参数属性所在标签位置：（12，6，6，0）
数据业务属性格式记录：（“数据业务”，“：”）
数据业务属性所在标签位置：（14，7，7，0）
摄像头属性格式记录：（“摄像头”，“：”）
摄像头属性所在标签位置：（16，8，8，0）

3.3.5 主题信息抽取

主题信息抽取模块的目的是根据规则库中的规则，抽取页面中的主题信息。其具体步骤如下：

1、 对输入的页面进行预处理，根据图 3-5 所示算法生成页面标签序列、内容向量。并根据页面链接地址得出该页面所在网站的域名。
2、 根据第一步生成的网站域名返回规则库中包含该域名的所有规则，将规则中的主题区域标签序列和第一步中生成的页面标签序列对比，生成一组待抽取区域，这组待抽取区域构成待抽取区域集合，待抽取区域集合生成过程见图 3-6 所示算法。
3、 判断第二步中生成的待抽取区域集合是否为空，如果为空，则说明页面标签序列中不包含主题区域标签序列，此时转新属性发现模块；否则转4。
4、 对待抽取区域集合中每一个待抽取区域，根据第二步返回规则中的位置向量和主题属性格式文件，判断是否可以抽取该区域中的信息，如果不能抽取该区域中信息，则转新属性发现模块；否则依据图 3-7 所示算法抽取该区域的主题信息，并返回主题信息向量。
5、 对规则中主题属性格式文件的每一个页面属性格式记录，通过属性清洗过程生成伪属性名，并将其映射到标准属性名，然后将第四步返回的主题信息向量中的信息根据标准属性名存入主题信息库中，页面的主题信息存储过程如图 3-8 所示算法。

```
输入：
页面标签序列：tags
内容向量：contents
规则库中规则：R
输出：
待抽取区域集合：As
GenRegions(tags,contents){
1、start=0,end=0,index=0,As=null;
2、While(true)
        //返回 tags 中第一个与 R.tags 相匹配子串的位置
        index=tags.IndexOf(R.tags);
        IF index<0 break;//如果没有找到与 R.tags 相匹配子串，终止循环
        //tmpStr 为 tags 中 0 到 index 的子串
        tmpStr=tags.subString(0,index);
       //求出待抽取区域第一个标签位置，numOfTags 计算 tmpStr 中标签的个数
        start=start+numOfTags(tmpStr);
        end= start+ numOfTags (R.tags);//求出 R.tags 中标签的个数
        As .add(R.tags,start,end);//将(R.tags,start,end)向量存入集合 As 中
        //将 tags 替换为 index+R.tags.length 位置后的子串
          tags=tags.subString(index+R.tags.length);
    EndWhile
3、return As;
}
```

图 3-6 页面待抽取区域集合生成算法

```
输入：
待抽取区域：A
规则：R
主题属性格式文件：F
内容向量：Contents
输出：
主题信息向量：Info
ExtracInfo(A,R,F,Contents){
1、num=numOfTags(A.tags),i=0;
2、While(i++<num)
```

```
            IF R.pos.p.isNoise==1//如果该标签内容是噪声，则进入下一个循环
            R.pos++;//规则 R 中位置向量向前推进一个元素
            continue;
        EndIF
        IF R.pos.p.attrType>0//如果该标签内容包含属性名
          //返回主题属性格式文件中对应属性格式记录中属性名
          str=F[R.pos.p.attrType].attrName;
          //判断该标签内容是否包含属性格式记录中属性名
          index=Contents[A.start+i].IndexOf(str);
          //如果不包含属性格式记录中属性名，则返回，该区域不再做判断
          IF index<0 return ;
          //如果标签内容即包含属性名又包含属性值
          IF R.pos. contentType >0
            //判断该标签内容是否包含属性格式记录中属性名
            index=Cotents[A.start+i].IndexOf(str);
            //如果不包含属性格式记录中属性名，则返回，该区域不再做判断
            IF index<0 return;
            str=Contents[A.start+i].subString(index+1);//返回属性值
            //将(R.pos.contentType,str)向量存入 Info 中
            Info.add(R.pos.contentType,str);
          EndIF
          R.pos++;
          contine;// 进入下一个循环
        EndIF
        IF R.pos.p. contentType >0//如果该标签内容仅为属性值。
          str=Contents[A.start+i];
          //将(R.pos.contentType,str)向量存入 Info 中
          Info.add(R.pos.contentType,str);
          R.pos++;
        EndIF
    EndWhile
3、return Info;
}
```

图 3-7 待选择区域主题信息抽取算法

```
输入：
主题信息向量：info
规则：R
StoreToResources(info,R){
    //res 为哈希向量，用于存储主题属性的属性名和属性值
1、len=info.size,j=0,res=null;
2、While(j++<len)
       //取得页面属性格式记录中属性名。
       attrName=R.F[info[j]. contentType]. attrName;
       //取得页面属性格式记录中属性名格式特征
       attrFeature=R.F[info[j]. contentType].attrFeature;
      //将页面中显示的属性名去掉属性名格式特征，生成伪属性名
      attrName=attrName.replaceAll(attrFeature);
      stand_attrName= f(attrName);//将伪属性名映射为标准属性名
      //如果 res 向量中有属性名为 stand_attrName 的属性值，则将新的内容合并到它原来的
      属性值中
      IF res.containAttribute(stand_attrName)
         res[stand_attrName].content+= info[j].str;
       EndIF
       ELSE//否则将主题属性值和属性名存入 res 中
          res .add(info[j].str, stand_attrName);
       EndELSE
    EndWhile
3、Resources.add(res);//将 res 向量中的元素存入 Resources 主题信息库中。
}
```

图 3-8 主题信息存储算法

3.4 实验

3.4.1 实验过程

为了验证本章方法的可行性，我们从五个有代表性的网站选择页面做实验，并以手机参数为主题。本实验的目的是验证利用本章方法能否准确、有效的抽取页面主题信息。

本章实验采用三个评价指标，分别是召回率、准确率和页面平均处理时间，

其描述形式如下。

- 召回率=$\frac{\text{正确抽取网页数}}{\text{应该抽取网页数}}$ 公式（3-2）

- 准确率=$\frac{\text{正确抽取网页数}}{\text{抽取所有网页数}}$ 公式（3-3）

- 页面平均处理时间=$\frac{\sum_i^n t_i}{n}$ 公式（3-4）

其中 n 表示某一网站样本页面中正确抽取的网页个数，t_i 为处理一个页面的时间。本文利用准确率和召回率反应本章方法抽取网页信息的正确性和完整性，利用页面平均处理时间反应本章方法处理页面的速度。

实验环境：Celeron CPU 2.4GHZ，1G 内存，WinXp 系统，Eclipse 平台。

表 3-1 描述了选取网站的信息，第一列表示选取网站的标识符，第二列表示选取的网站域名，第三列表示对应网站选取的页面数。

表 3-1 选取网站信息描述

标识符	网站网址	网页数
W1	http://www.pcpop.com/	86
W2	http://www.pconline.com.cn/	110
W3	http://www.21cn.com	103
W4	http://www.163.com	103
W5	http://www.sohu.com	112

实验中样本页面是从各个网站抽取的包含手机参数主题信息的页面。综合各个网站的手机参数主题属性记录如下：

手机参数主题属性记录：（手机名，参考价格，上市时间，手机制式，手机外形，主屏参数，数据业务，摄像头，商家报价，主屏尺寸，主屏分辨率，主屏颜色，手机通话时间，摄像头像素，音乐播放，产品类别，生产厂商，待机时间，摄像像素）

伪属性名和标准属性名映射方式如表 3-2，第一和第三行表示标准属性名，第二和第四行表示相应的伪属性名，该表中省略了伪属性名和标准属性名一致的情况。

表 3-2 伪属性名和标准属性名映射方式

标准属性名	参考价格	上市时间	主屏参数	手机外形
伪属性名	参考价格、参考报价	上市时间、上市日期	主屏参数、屏幕参数	手机外形、外观设计
标准属性名	主屏尺寸	主屏分辨率	手机制式	
伪属性名	主屏尺寸 、尺寸体积	屏幕色彩、主屏分辨率	手机制式、网络制式、网络频率	

样本训练模块中通过样本生成规则中主题区域标签序列和对应网站每个样本页面包含的属性个数如表 3-3 所示，并对 W3 网站建立了两条主题区域标签序列，这是由于 W3 网站中有的页面包含了 9 个属性，有的页面只包含了 8 个属性，而只建立一条抽取规则时难已覆盖整个网站手机参数相关的页面，则本文对建立了该网站两条抽取规则。

表 3-3 样本页面规则中主题区域标签序列

标识符	主题区域标签序列	页面包含属性个数
W1	div*div*SPAN*a*SPAN*a*b*span*a*BR*SPAN*a*SPAN*a*b*b*span*a*BR*SPAN*A*BR*SPAN*A*BR*SPAN*BR*SPAN*A*BR*SPAN*A*BR*SPAN*A*BR*SPAN*BR*SPAN*A	11
W2	dt*dd*strong*span*dd*strong*dd*strong*dd*strong*dd*strong*dd*strong*dd*strong	8
W3	span*tr*td*tr*td*div*div*a*tr*td*div*div*a*tr*td*div*div*tr*td*div*div*tr*td*div*div*tr*td*div*div*tr*td*div*div*tr*td*div*div*tr*td*div*div	9
	span*tr*td*tr*td*div*div*a*tr*td*div*div*a*tr*td*div*div*tr*td*div*div*tr*td*div*div*tr*td*div*div*tr*td*div*div*tr*td*div*div	8
W4	h3*p*p*p*p*p*p*div	5
W5	td*tr*td*tr*td*strong*span*tr*td*tr*td*strong*tr*td*tr*td*strong*tr*td*tr*td*strong*tr*td*tr*td*strong*tr*td*tr*td*strong*tr*td*tr*td*strong*tr*td*tr*td*strong*tr*td*tr*td*strong	9

本章在 3.3.4 节中的例子已经对 W2 网站的页面属性格式记录和对应属性标签位置进行了详细说明。从表 3-3 可以看出，除了 W4 网站外，各个主题区域标

签序列都至少包含了 10 个以上的标签，限于篇幅原因，本章不再对其他网站由样本生成的规则中页面属性格式记录和位置向量做过多说明。

其综合实验结果如表 3-4 所示。

表 3-4 综合各个网站抽取实验结果

标识符	正确抽取网页数	抽取所有网页数	应当抽取网页数	准确率	召回率
W1	82	82	86	100%	95.3%
W2	110	110	110	100%	100%
W3	94	100	104	94%	90.3%
W4	103	103	103	100%	100%
W5	112	112	112	100%	100%

3.4.2 实验分析

从表 3-4 可以看出，W1 和 W3 的召回率分别为 95.3%和 90.3%，这是由于在 W1 和 W3 网站中均有 4 个页面没有被发现，通过分析我们发现这几个页面中包含的主题属性个数和样本中包含的属性个数不一致，在 W1 网站中，我们建立的规则包含了 11 个属性，而在这四个未发现的的页面中只包含了 10 个已建立的需要抽取的属性，由于规则中主题区域标签序列是按照属性的个数建立的，这样在这些网页建立的标签序列中找不到该网站规则中的主题区域标签序列，所以并未发现这些网页；同样在 W3 网站中，我们发现部分页面只有 7 个属性，而我们建立的规则只是考虑了包含 8 个和 9 个属性的情况，所以这些页面在实验中也未发现，这说明在满足准确抽取页面信息情况下，本文建立抽取规则中主题区域标签序列应当尽可能完善的反应主题信息特征。同时 W3 网站的准确率只有 94%，通过分析我们发现在 W3 网站中有 6 个页面只是抽取了页面中部分主题信息，这些页面中虽然通过本章建立的页面标签序列中找到了该网站规则中的主题区域标签序列，但是 W3 网站这些页面中出现了新属性 GPS 功能和电子字典，这样当使用规则中位置向量和主题属性格式文件进行页面主题信息抽取时，只是抽取以前已经存在属性的信息，而对于位置向量中某些存在主题信息的位置，由于该位置出现了新属性，所以在主题属性格式文件中找不到相应的页面属性格式记录，从而不能抽取出新属性的信息，这说明本章在样本中建立的规则还不能覆盖这两个网站所有包含手机信息主题的页面，如果以这些出错的页面为样本页面再建立抽取规则，则能准确的抽取出页面的信息，如在 W3 网站中我们相应的建立两条抽取规则，这样从一定程度上提高了该网站抽取页面信息的准确率和召回率。

但是本章的方法在 W2、W4 和 W5 网站中都能正确的抽取出网页中的手机参数主题信息。综合各个网站的抽取结果，我们发现利用本方法平均准确率和平均召回率比较高，这证明本章的方法具有一定的应用可行性；同时这也说明在抽取过程中，包含主题信息的页面通常都能准确的抽取出所需要的信息，所以图 3-4 所示模型中新属性发现模块的运行机率比较低。

在本章的方法中，页面处理时间分为四个部分，一是页面下载时间 Tdownload，二是页面预处理时间 Ttag，三是待抽取区域生成时间 Tarea，四是主题信息抽取时间 Textract，则一个页面处理时间 t_i =Tdownload+Ttag+Tarea+Textract，由于 Tdownload 受网络带宽等非主观性原因的影响，本章设定一个页面处理时间为： t_i = Ttag+Tarea+Textract。

目前很多方法都是通过对网页建立 DOM 树，然后分析 DOM 树，抽取页面中的主题信息，文献[9][10][11][12][13][14]都属于这种方法。该方法的页面处理时间分为三个部分，一是页面下载时间 Tdownload，二是创建 DOM 树的时间 Tdom，三是分析 DOM 树抽取页面信息时间 Tdextract。由于抽取页面信息采用的策略不同，该方法中 Tdextract 的值一般不相同，而 Tdownload 受网络影响较大，但是它们建立 DOM 树的时间 Tdom 却基本相同，本文通过 Cobra 开源软件生成各个网站页面平均建立DOM树的时间与本章方法页面平均处理时间进行比较，其结果如表 3-5 所示。

表 3-5 页面处理时间比较

标识符	本方法页面平均处理时间 ms	页面平均建立 DOM 树时间 ms	网页数
W1	8492.5	11968.5	10
W2	422.6	2986.7	10
W3	449.3	11754.2	10
W4	437.4	7197.1	10
W5	3380.8	7740.6	10

从表 3-4 和 3-5 可以看出，采用 DOM 树的方法仅创建 DOM 的时间就远大于本方法中包含信息抽取的页面处理时间。因此，本章采用的方法在满足高准确率和召回率的前提下，相对于分析 DOM 树抽取页面信息的方法从一定程度上减少页面处理时间，这在搜索引擎的信息采集应用中有实际的应用价值。

3.5 本章小结

本章主要设计了一种基于标签序列的 Web 页面主题信息抽取方法，并建立该方法相应的模型。该方法通过一定策略，生成页面标签序列，并以此建立抽取

规则，通过规则抽取页面中的主题信息，一定程度上降低了抽取 Web 页面主题信息的复杂性，并减少了处理 Web 页面的时间。实验中将本方法应用到部分网站页面中的手机参数主题信息抽取中，实验结果表明应用本方法抽取各个网站页面的召回率和准确率都比较高，具有较好的应用可行性。

第四章 基于可信度的Web页面主题新属性发现

4.1 引言

目前，分装器（Wrapper）已经广泛应用于Web信息抽取中，但是当数据源结构发生变化，已建立的分装器将失去抽取数据源数据的功能。针对这一弱点，研究人员提出了很多的解决办法，如通过训练样本或者已抽取的数据，对样本或者已抽取的数据进行分类[35][50]，当出现新结构的页面时，将新结构页面和训练样本等进行比较，判断页面是否需要抽取，进而归纳出新的抽取规则，使一些分装器具有了自主学习的方法，这从一定程度上降低了人工参与。但是如果在新结构页面中出现的有用信息不能在训练样本等中找到分类时，分装器将无法判断是否需要抽取页面中信息，则此时有用信息被丢弃。

考虑下面情况，已建好的分装器中规则能正确抽取如下表单中手机名、参考价格两个属于手机参数信息的属性。

```
<table>
    <tr>
        <td>手机名：</td><td>诺基亚 1200</td>
    </tr>
    <tr>
        <td>参考价格：</td><td>400</td>
    </tr>
</table>
```

当新出现的页面结构如下：

```
<table>
    <tr>
        <td>手机名：</td><td>诺基亚</td>
    </tr>
    <tr>
        <td>参考价格：</td><td>400</td>
    </tr>
    <tr>
        <td>手机制式：</td><td>GSM</td>
    </tr>
```

</table>

在新的页面结构下出现的主题属性手机制式在训练样本等中并没有出现过，将不能准确找到它的分类，这样手机制式属性值将不会被抽取，从而丢失了有用信息，如文献[31][32][33][34][35]建立的分装器都不能有效解决这个问题，采用本文第三章的方法建立的分装器也属于这种情形，如本文第三章实验中W3标识的网站页面中，由于出现了新的属性电子字典和GPS，所以该网站抽取页面信息的准确率受到了一定的影响。

为解决上述问题，文献[48]通过贝叶斯学习方法和EM算法建立一个概率框架来发现这些新属性，但是该方法中实现过程太过复杂，并且概率框架中部分参数分布难以量化，而且EM算法本身有一些固定的缺点，如EM算法的收敛速度比较慢等。

基于上述原因本文引入可信度理论，通过对待抽取属性和已抽取属性的特征进行分析，定义一系列的证据和规则，提出一种基于可信度的Web页面主题新属性发现方法，以判断待抽取属性是否为主题新属性。该方法降低了文献[48]方法的复杂度，并选取部分网站手机参数页面实验，取得了较好的效果。通过本章的方法，能较准确发现Web页面主题的新属性，从而能修改分装器中的规则，完善分装器的功能。

4.2 可信度

可信度[60]（CF(H,E)）是根据经验对一个事物或者现象为真的相信程度, 为了更好的理解可信度，研究人员提出了如下定义：

CF(H,E)=MB(H,E)-MD(H,E) 公式 （4-1）

上式中MB(measure belief)为信任增加长度，表示因证据E的出现而增加了现象H为真的信任增加程度，当P(H,E)>P(H)时，MB(H,E)>0。MD(measure disbelief)为不信任增加长度，表示因证据E的出现对现象H为假的信任增加程度，当P(H,E)<P(H)时，MD(H,E)>0，其中P(H|E)表示在证据E出现的情况下出现现象H的概率，P(H)表示现象H出现的概率。MB和MD定义如下：

$$\mathrm{MB(H,E)}=\begin{cases}\dfrac{\max\{P(H\mid E),P(H)\}-P(H)}{1-P(H)}, & \text{当}P(H)\neq 1\\ 1, & \text{当}\mathrm{P}(H)=1\end{cases} \qquad \text{公式 （4-2）}$$

$$\mathrm{MD(H,E)}=\begin{cases}\dfrac{\min\{P(H\mid E),P(H)\}-P(H)}{1-P(H)}, & \text{当}P(H)\neq 0\\ 1, & \text{当}\mathrm{P}(H)=0\end{cases} \qquad \text{公式（4-3）}$$

结合公式（4-1）、（4-2）和（4-3），则有：

$$CF(H,E)=\begin{cases}\dfrac{p(H\mid E)-p(H)}{1-p(H)}, & 当p(H\mid E)\neq p(H)\\ 0, & 当p(H\mid E)=p(H)\end{cases}$$ 公式 （4-4）

其中$-1\le CF(H,E)\le 1$，当$CF(H,E)>0$时，表示证据E增加了结论为真的程度。

当有多个证据E_1，E_2，E_3，……，E_n同时出现时，现象H为真可信度定义如下：

$$CF(H,E)= a_1* CF(H,E_1) + \dots\dots + a_n* CF(H,E_n)$$ 公式 （4-5）

其中a_i为每个证据对现象H支持的权重，由专家事先给出，且$a_1+\dots\dots+a_i+\dots\dots+a_n=1(0\le a_i\le 1)$。

4.3 一种基于可信度的 Web 页面主题新属性发现方法

在本文的方法中，我们将可信度应用于页面主题新属性的发现。通过定义待抽取属性需要抽取的证据，建立了该方法相应的模型，通过模型对该方法实现方式进行详细的阐述。在模型中我们通过定义各个证据相应的规则，量化各个证据的可信度，从而实现判断待抽取属性是否为主题的新属性。在页面源代码中属性可以由属性名和属性名格式特征表示，本文主要考虑的是属性名的发现，进而确定是否发现页面中主题的新属性。

4.3.1 证据定义

由于 DOM（Document Object Model）树能够很好的反映 Web 页面结构信息，目前已有很多的研究人员将 DOM 树应用到 Web 信息抽取中，本文采用开源软件 Cobra 对输入的 Web 页面先生成 DOM 树。在 DOM 树中每个待抽取属性都属于树中叶子节点，与文献[14][56]类似，本文定义 DOM 树的叶子节点为一个基本对象。根据 DOM 树的特点以及主题属性在页面中显示格式的特点，本文建立如下五条证据。

证据 1：待抽取属性名字体与已抽取属性名字体之间关系影响程度（F）

该证据主要是判断待抽取属性名字体与已抽取属性名字体是否相同，从而决定待抽取属性需要抽取的影响程度。

证据 2：待抽取属性名背景与已抽取属性名背景之间关系影响程度（B）

通过 3.2 节中页面结构分析，我们知道页面的主题信息通常拥有视觉区域连续性，则在该区域中的内容通常具有相同的背景特征，因此判断待抽取属性名背景与已抽取属性名背景是否相同，从而决定待抽取属性需要抽取的影响程度。

证据3：待抽取属性名与已抽取属性名拥有相同父节点影响程度（P）

通过3.2节中页面结构特征分析，我们知道页面的主题信息通常在页面源代码中的一个连续区域出现，则根据Web页面生成的DOM树，各种主题的属性在页面中可能会有相同的父节点，基于此假设，本文认为待抽取属性名与已抽取属性名是否拥有相同的父节点对待抽取属性是否需要抽取有影响。

证据4：待抽取属性名与已抽取属性名格式相同影响程度（C）

根据本文第三章3.2节中主题属性页面显示格式的分析，我们知道同一主题不同的属性在页面中可能有相同的属性名格式特征，基于此假设，本文认为待抽取属性名与已抽取属性名格式是否相同对待抽取属性是否需要抽取有影响。

证据 5：待抽取属性名与用户感兴趣范围影响程度（S），即待抽取属性名是否属于用户感兴趣信息的程度。

Web 信息抽取是通过计算机从大量的 Web 数据中抽取感兴趣的信息，待抽取属性是否为主题的新属性取决于用户是否对此数据感兴趣。

4.3.2 基于可信度的 Web 页面主题新属性发现模型

根据本章4.3.1节定义的证据，相应的对每一个证据都有一个可信度，分别是字体关系可信度(CF(H,F))、背景关系可信度(CF(H,B))、待抽取属性名与已抽取属性名拥有相同父节点可信度(CF(H,P))、待抽取属性名与已抽取属性名格式相同可信度(CF(H,C))和待抽取属性名与用户感兴趣范围关系可信度(CF(H,S))，再根据这些可信度最终确定待抽取属性需要抽取的可信度CF(H,E)，并建立了基于可信度Web页面主题新属性发现方法的模型如图4-1所示：

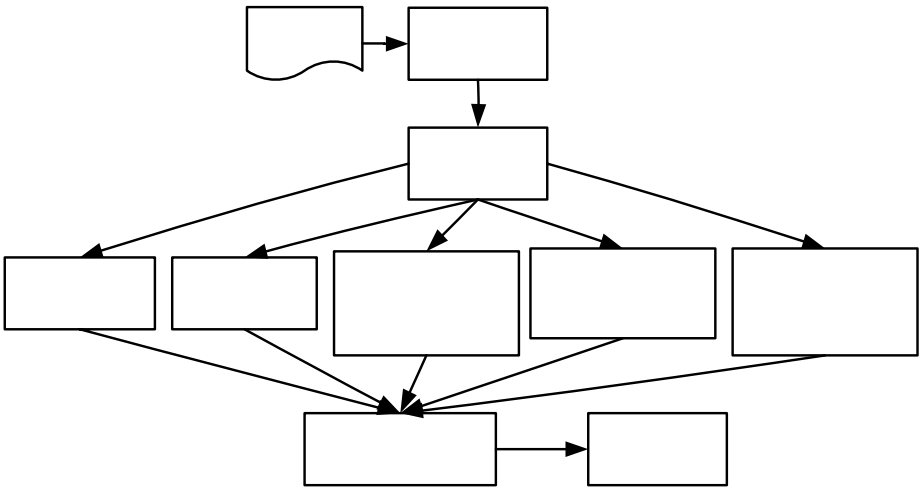

图4-1 基于可信度的Web页面主题新属性发现模型

从图4-1可以看出，该模型主要是本章定义的证据可信度的确定。根据公式（4-5），可以得出待抽取属性需要抽取的可信度如公式（4-6）。根据公式（4-6）确定待抽取属性需要抽取的可信度CF(H,E)，由CF(H,E)的值决定待抽取属性是否

为主题的新属性。

$$CF(H,E)=a_1* CF(H,F)+a_2* CF(H,B)+a_3* CF(H,P)+a_4* CF(H,C)+a_5* CF(H,S)$$ 公式 （4-6）

其中CF(H,E) 为待抽取属性需要抽取的可信度，H为待抽取属性需要抽取现象。

该模型的主要功能是发现页面中主题的新属性，其具体步骤如下：

1、对输入页面建立相应的DOM树T，层次遍历DOM树生成待抽取属性集合S。

2、顺序从S中选择一个待抽取属性，结合生成的DOM树T，确定CF(H,F)、CF(H,B)、CF(H,P)、CF(H,C)和 CF(H,S)。

3、综合第2步骤中的各个证据的可信度，根据公式（4-6）确定待抽取属性需要抽取的可信度C(H,E)。当CF(H,E) > Threshold（Threshold为用户自定义阀值）时，认为发现主题的新属性，否则认为没有发现主题的新属性，转4.

4、判断S集合中是否还有待抽取属性，有则转2，否则退出程序。

由于在本章开始时，本文假设可以通过分装器定义的规则抽取页面中部分主题属性及其内容，则本文设定已抽取主题属性的各个特征都能事先确定。

在实际应用中，公式（4-4）中的 P(H|E)和 P(H)值通常难以获得，因而 CF(H,E)的值一般由领域专家直接给出，本文根据 4.3.1 节定义的证据，设定一系列规则，量化 P(H|E)。并认为待抽取属性需要抽取的可能性和不可能性是相同的，因此一般令 P(H)=0.5，再根据公式（4-4），从而可以求出 CF(H,E)。

以下分别对各个证据的可信度进行具体介绍。

4.3.3 字体关系可信度

由于主题属性名在页面源代码中表现为一个字符串，它的字体是由它的祖先节点的 HTML 标签和其属性决定的，而 HTML 标签是一个确定的结合，我们将 HTML 标签分为两类，一类是和字体有关的标签集合，该类标签 fonts={"h1","h2","h3","h4","h5","h6","b","i","u","tt","sup","sub","s","strike","em", "strong","code","samp","kdb","var","dfn","cite","small","big","b",CSS}，其中 CSS 表示页面中 CSS 样式中表示的字体；所有不属于这个集合中的 HTML 标签都属于第二类标签。则对应的主题属性名有两种字体形式，一是默认字体，即在主题属性名所在叶子节点的祖先节点 HTML 标签中找不到集合 fonts 中元素，此时我们将主题属性名字体表示为一个空字符串；二是非默认字体，即在主题属性名所在叶子节点的祖先节点 HTML 标签中找到了集合 fonts 中元素，此时需要找到该

祖先节点，再根据 DOM 树节点的特点，将 DOM 树节点的标签名、属性和属性值拼接为一个字符串，我们将这个过程称为属性字体特征化过程，将这个字符串称为字体特征串。如图 4-2 中参考价格属性名字体经过属性字体特征化过程后字体特征串为“strong”，主屏尺寸属性名字体经过属性字体特征化过程后字体特征串为“fontsize=5”，摄像头属性名经过属性字体特征化过程后为“font1”。这样待抽取属性名和已抽取属性名字体的比较就转化为两个字符串的比较。根据以上分析，并定义待抽取属性名字体与已抽取属性名字体之间关系影响程度规则如表 4-1 所示。

表4-1 字体关系影响程度规则

序号	影响结构因素	P(H|F)		
		相同	不完全相同	不相同
1	待抽取属性名与已抽取属性名的字体关系	1	0.5	0

由于主题通常包含多个属性，则已抽取属性名的字体特征串将形成一个集合，本文假设它为 FS，待抽取属性名特征字体特征串为 fs，该规则的具体解释如下：

1、如果 fs$\notin$ FS，则 P(H|F)=0；

2、如果 fs$\in$ FS 且 fs=””，则 P(H|F)=0.5；

3、如果 fs$\in$ FS 且 fs$\neq$ ””，则 P(H|F)=1。

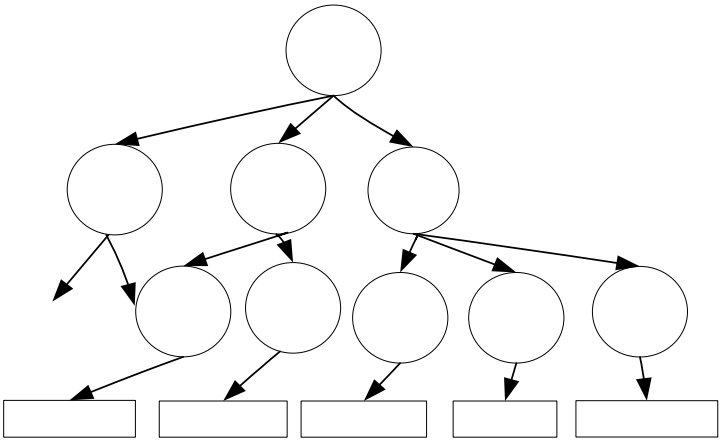

图 4-2 DOM 树中特征示例

结合公式（4-4）和表 4-1 的规则，则可以求出 CF(H,F)。图 4-3 所示算法描述了待抽取属性名字体与已抽取属性名字体之间关系影响程度规则的算法。

输入：

已抽取属性的字体特征串集合：FS

字体标签集合：fonts

待抽取属性在DOM树中的节点：attrNode

输出：

P(H|F)的值：R，R∈ {0,0.5,1}

算法描述：

```
Getphf(FS, fonts,attrNode){
      //取得attrNode的父节点
  1、R=0,parent=attrNode.getParentNode();
  2、While(parent!=null)
          str="";
        //生成字体征串,其中getAttributeName方法用于返回标签中某一
        //属性的属性名，getAttributeValue用于返回该属性的属性值
          IF parent.getNodeName
            While(parent.getAttribute) //  对于parent节点标签中的每一个属性
              IF parent.getAttributeName()!=null
                &&parent.getAttributeValue()!=null
               str=str+parent.getAttributeName()+"="+parent.getAttributeValue();
              EndIF
            EndWhile
            IF str∈ FS//判断父节点标签名是否在S中
              R=1;
              转下一个待抽取属性处理。Break;
            EndIF
          EndIF
        //如果parent所在节点标签的CSS样式中属于fonts
        IF parent. getAttributeName (CSS)
          IF parent.getAttributeValue(CSS)∈ fonts；
            str= parent.getAttributeValue(CSS);
          EndIF
        EndIF
        //取得parent的父节点
        parent=parent.getParentNode();
```

```
        EndWhile
        IF str=""且FS中有已抽取属性名的字体特征串=""
            R=0.5;
        EndIF
    3、return R;
}
```

图 4-3 待抽取属性名字体与已抽取属性名字体关系规则算法

4.3.4 背景关系可信度

背景关系可信度的确定和字体关系可信度确定相似，页面区域的背景主要由属性所在叶子节点的祖先节点HTML标签的属性决定，其主要决定页面区域背景的属性有bks={“bg”,”bgcolor”,”background”,CSS}，其中CSS表示页面中CSS样式中表示的背景。和字体关系可信度确定相似，我们定义两中背景形式，一种是默认背景，即在主题属性名所在叶子节点的祖先节点HTML标签属性中找不到集合bks中元素，此时我们将主题属性背景表示为一个空字符串；二是非默认背景，即在主题属性名所在叶子节点的祖先节点HTML标签属性中找到了集合bks中元素，此时找到该祖先节点，再根据DOM树节点的特点，将DOM树节点标签的属性和属性值拼接为一个字符串，我们将这个过程称为属性背景特征化过程，将这个字符串称为背景特征串，如图4-2中主屏尺寸属性名背景经过属性背景特征化过程后背景特征串为“bgcolor=#ffffff”，参考价格属性名背景经过属性背景特征化过程后为“class=bg1”，根据以上分析，定义待抽取属性名背景与已抽取属性名背景之间关系影响程度规则如表4-2所示。

表4-2 背景关系影响程度规则

序号	影响结构因素	P(H|B)		
		相同	不完全相同	不相同
1	待抽取属性名与已抽取属性名的背景关系	1	0.5	0

由于主题通常包含多个属性，则已抽取属性名的背景特征串将形成一个集合，本文假设它为 BS，待抽取属性特征背景特征串为 bs，该规则的具体解释如下：

1、如果 bs$\notin$ BS，则 P(H|B)=0；

2、如果 bs$\in$ BS 且 bs=””，则 P(H|B)=0.5；

3、如果bs$\in$ BS且bs$\neq$ ””，则P(H|B)=1。

结合公式（4-4）和表4-2的规则，则可以求出CF(H,B)。图4-4所示算法描述了待抽取属性名背景与已抽取属性名背景之间关系影响程度规则的算法。

```
输入：
背景特征串集合：BS
待抽取属性在DOM树中的节点：attrNode
背景属性集合bks
输出：
P(H|B)的值：R，R∈ {0,0.5,1}
Getphb(BS,attrNode,bks){
      //取得attrNode的父节点
  1、R=0,parent=attrNode.getParentNode();
  2、While(parent!=null)
        str="";
       IF parent. getAttributeName∈ bks
       IF parent.getNodeName∈ bks
          //查找parent节点中是否含有名为bg的属性
          IF parent.getAttributeName("bg")!=null
             //生成背景特征串，其中parent.getAttributeValue("bg")为bg
             //属性的属性值
              str="bg="+parent.getAttributeValue("bg");
             break;
          EndIF
          IF parent.getAttributeName("bgcolor")!=null//同分析bg属性类似
              str="bgcolor="+parent.getAttributeValue("bgcolor");
              break;
          EndIF
          IF parent.getAttributeName("background")!=null//同分析bg属性类似
              str= "background="+parent.getAttributeValue("background");
              break;
          EndIF
       EndIF
       //如果parent所在节点标签的CSS样式中属于bks
       IF parent. getAttributeName(CSS)
          IF parent. getAttributeValue (CSS) ∈ bks
```

```
                str= parent. getAttributeValue (CSS);
            EndIF
        EndIF
        parent=parent.getParentNode();//取得parent的父节点
    EndWhile
    IF str∈ BS
        IF str!=""
            R=1;
        EndIF
        ELSE
            R=0.5;
        EndELSE
    EndIF
  3、return R;
}
```

图 4-4 待抽取属性名背景与已抽取属性名背景关系规则算法

4.3.5 待抽取属性名与已抽取属性名拥有相同父节点可信度

由于主题属性名为DOM树中叶子节点，则待抽取属性名与已抽取属性名拥有相同的父节点可信度确定主要面临两个问题，一是已抽取属性名父节点集合PS的确定；二是如何判断待抽取属性名父节点属于已抽取属性名父节点集合PS。对于第一个问题，由于我们已经有了已抽取属性的部分特征，则我们能确定这些已抽取属性名最近的祖先节点ancestor，我们认为最近的这个祖先节点构成已抽取属性名父节点集合PS；根据祖先节点和已抽取属性名所在节点我们能确定它们之间的路径长度L，根据路径长度我们来判断待抽取属性名父节点是否属于已抽取属性父节点集合，其具体判断步骤如下：

1、令i=0，L为路径长度，已抽取属性名父节点集合PS，待抽取属性名所在节点ps。

2、获取ps所在节点的父节点parent，ps=parent，i++。

3、如果i小于L，转2；否则，查找PS中是否有ps节点，如果找到则认为待抽取属性名父节点属于已抽取属性名父节点集合PS，否则认为待抽取属性名父节点不属于已抽取属性名父节点集合PS。

如在图4-2中，假设已抽取属性为：参考价格、主屏尺寸和摄像头。则主屏尺寸和摄像头属性名拥有的最近的祖先节点为tbody，从属性名所在节点到tbody

节点的路径长度为2，参考价格、主屏尺寸和摄像头拥有的最近的祖先节点为form，从属性名所在节点到form节点的路径长度为3。如果出现新的属性数据业务，通过查找数据业务属性名所在叶子节点的祖先节点，发现form为其祖先节点，且form节点到数据业务属性名所在叶子节点的路径长度为2，则我们认为数据业务属性名父节点属于已抽取属性父节点集合。

据此本章定义待抽取属性名与已抽取属性名拥有相同的父节点对待抽取属性是否需要抽取影响程度规则如表4-3所示。

表4-3 待抽取属性名与已抽取属性名拥有相同的父节点影响程度规则

序号	影响结构因素	P(H|P)	
		属于	不属于
1	待抽取属性名与已抽取属性名拥有相同的父节点	1	0

由于使用Cobra对网页建立DOM树时，该树的根节点总为#document，则ps不可能为空。结合公式（4-4）和表4-3的规则，则可以求出CF(H,P)，。

4.3.6 待抽取属性名与已抽取属性名格式相同可信度

待抽取属性名与已抽取属性名格式相同可信度的关键问题是怎样确定属性的格式，同一主题不同属性，在页面上的显示格式并不相同，主要是由属性格式特征不同造成的，本文主要考虑属性名格式特征的影响，我们定义仿照本文第三章属性格式记录定义，定义本章中某一个属性的属性名格式记录为：

(attrName,attrFeature,attrTagName)

其中attrName为属性名，attrFeature为属性名格式特征，attrTagName表示环绕该属性名字符串的标签名。

由于DOM树中每个待抽取属性名都属于树中的叶子节点，它也是一个字符串，attrFeature表示该叶子节点所在字符串所含有的特殊字符集合，如图4-2中参考价格属性的属性名格式记录为(“参考价格”,”：”,”strong”)。

由于一个主题有多个属性，已抽取属性名可以形成一个属性名格式记录集合CS，判断待抽取属性名所在叶子节点字符串cs中是否包含attrFeature和环绕该属性名字符串的标签名ct是否和环绕已抽取属性名的标签名attrTagName相同，本章定义待抽取属性名与已抽取属性名格式相同影响程度规则如表4-4所示。

表4-4 待抽取属性与已抽取属性格式相同影响程度规则

序号	影响结构因素	P(H|C)		
		包含	不完全包含	不包含
1	待抽取属性与已抽取属性格式相同	1	0.5	0

该规则的具体解释如下：

1、当CS[i]. attrFeature∈ cs&& CS[i]. attrTagName=ct&&CS[i]. attrFeature≠””，则P(H|C)=1；

2、当CS[i]. attrFeature∈ cs&& CS[i]. attrTagName=ct&&CS[i]. attrFeature=””，则P(H|C)=0.5；

3、当CS[i]. attrFeature∉ cs|| CS[i]. attrTagName∉ ct，则P(H|C)=0。

其中CS[i]. attrFeature表示主题中已抽取的第i个属性的属性名格式特征，CS[i]. attrTagName表示环绕第i个已抽取属性属性名的标签名。结合公式4-4和表4-4的规则，则可以求出CF(H,C)。

4.3.7 待抽取属性名与用户感兴趣范围关系可信度

待抽取属性名与用户感兴趣范围关系可信度确定主要是判断待抽取属性名as是否属于用户感兴趣范围，则如何确定用户感兴趣范围是该步骤的关键，本文采用的方法是从多个网站收集主题属性名，由多个属性名的集合构成用户感兴趣范围As。

该步骤的意义是对于同一主题，分装器对于不同的页面有不同的规则，抽取的主题属性并不一致，即可能出现这种情形，A页面中包含主题属性attrA，但是B页面中在分装器规则建立时，并没有attrA，但是后来调整了B页面的部分结构，出现了attrA，则B页面中的attrA属性也应当抽取出来，本章定义待抽取属性名与用户感兴趣范围关系规则如表4-5所示。

表4-5 待抽取属性名与用户感兴趣范围关系规则

序号	影响结构因素	P(H|S)	
		属于	不属于
1	待抽取属性名属于用户感兴趣范围	1	0

该规则的具体解释如下：

1、当as∈ AS，则P(H|S)=1；

2、当as∉ AS，则P(H|S)=0。

结合公式（4-4）和表4-5的规则，则可以求出CF(H,S)。

4.4 实验

4.4.1 实验过程

本章以部分网站的手机参数页面为样本进行实验，验证本章提出的方法可否正确发现页面中的主题新属性。本章实验采用两个评价指标，其描述形式如下：

- 召回率=$\frac{\text{正确发现属性个数}}{\text{应当发现属性个数}}$ （公式4-7）

- 准确率=$\frac{\text{正确发现属性个数}}{\text{发现所有属性个数}}$ （公式4-8）

表4-6描述了选取网站信息，第一列表示选取网站的标识符，第二列表示选取的网站域名，第三列表示对应网站选取的属性个数。

表4-6 选取网站信息描述

标识符	网站网址	属性个数
W1	http://www.pcpop.com/	10
W2	http://www.pconline.com.cn/	7
W3	http://www.21cn.com	9
W4	http://www.zol.com.cn	8

由于Web页面内容的主体部分在标签<body></body>之间，本文只选择页面源代码中<body></body>标签之间内容进行分析。

W1和W4网站手机参数信息包含了参考价格和手机制式两个属性，W2网站包含了参考价格和网络制式两个属性，W3网站包含了参考报价和手机制式两个属性，由第三章3.4节我们知道参考报价为手机参数主题参考价格属性，网络制式为手机参数主题手机制式属性，则我们开始建立的分装器中规则只是抽取这两个属性。两个属性的初始条件如表4-7至4-9所示。

各个网站手参考价格、手机制式属性名的字体特征串如表4-7所示。

表4-7 各个网站属性名字体特征串

标识符	参考价格	手机制式
W1	“font12-blue-bold”	“font12-blue-bold”
W2	“strong”	“strong”
W3	“”	“”
W4	“Cl_31 hui612”	“Cl_31 hui612”

各个网站参考价格、手机制式属性名的背景特征串如表4-8所示。

表4-8 各个网站属性名背景特征串

标识符	参考价格	手机制式
W1	“”	“”
W2	“class=info_con”	“class=info_con”
W3	“”	“”
W4	“”	“”

由于各个网站主题属性的父节点为DOM树中一个节点，它在程序运行过程中动态生成，本文在此不做详细描述。

各个网站手机名称、参考价格属性的属性名格式记录如表4-9所示。

表4-9 各个网站属性名格式记录

标识符	参考价格	手机制式
W1	(“参考价格”,“”,”a”)	(“手机制式”,””,”A”)
W2	(“参考价格”,“：”,”strong”)	(“网络制式”,”：”,”strong”)
W3	(“参考报价”,“：”,”div”)	(“手机制式”,”：”,”div”)
W4	(“参考价格”,“”,”a”)	(“手机制式”,””,”a”)

对于手机参数主题，用户感兴趣范围是本文通过抽取部分网站手机参数名形成的一个文件，它实际上相当于一个字典，限于本文篇幅，在此不做详细说明。

由于事先并没有对每个证据支持的权重a_i做实验，而a_i可以在大量的网站页面中实验获得，本文假设各个证据对待抽取属性需要抽取的影响程度相同，则此时令公式（4-6）中$a_1=a_2=a_3=a_4=a_5=1/5$。

W1、W2、W3、W4网站页面各个选择的属性对应证据的可信度如表4-10到4-13所示：

表4-10 W1网站页面属性对应证据可信度

属性	商家报价	主屏尺寸	主屏颜色	手机通话时间	摄像头像素	音乐播放	主屏分辨率	产品尺寸
CF(H,F)	1	1	1	1	1	1	1	1
CF(H,B)	0	0	0	0	0	0	0	0
CF(H,P)	1	1	1	1	1	1	-1	-1
CF(H,C)	0	0	0	0	0	0	-1	-1
CF(H,S)	1	1	1	1	1	1	1	1
CF(H,E)	0.6	0.6	0.6	0.6	0.6	0.6	0	0

表4-11 W2网站页面属性对应证据的可信度

属性	上市时间	手机外形	主屏参数	数据业务	摄像头
CF(H,F)	1	1	1	1	1
CF(H,B)	1	1	1	1	1
CF(H,P)	1	1	1	1	1
CF(H,C)	1	1	1	1	1
CF(H,S)	1	1	1	1	1
CF(H,E)	1	1	1	1	1

表4-12 W3网站页面属性对应证据的可信度

属性	产品类别	生产厂商	上市日期	手机类型	屏幕色彩	主屏尺寸	电子字典
CF(H,F)	0	0	0	0	0	0	0
CF(H,B)	0	0	0	0	0	0	0
CF(H,P)	1	1	1	1	1	1	1
CF(H,C)	1	1	1	1	1	1	1
CF(H,S)	1	1	1	1	1	1	-1
CF(H,E)	0.6	0.6	0.6	0.6	0.6	0.6	0.2

表4-13 W4网站页面属性对应证据的可信度

属性	商家报价	上市日期	手机类型	屏幕色彩	和弦铃声	摄像头
CF(H,F)	1	-1	1	1	1	1
CF(H,B)	0	0	0	0	0	0
CF(H,P)	1	1	1	1	1	1
CF(H,C)	0	-1	0	0	0	0
CF(H,S)	1	1	1	1	1	1
CF(H,E)	0.6	0	0.6	0.6	0.6	0.6

当Threshold=0时，综合表4-10到4-13，则各个网站页面的实验结果如表4-14所示。

表4-14 Web页面主题新属性发现综合实验结果

标识符	新发现的属性	未发现的新属性	准确率	准确率
W1	商家报价、主屏尺寸、主屏颜色、手机通话时间、摄像头像素、音乐播放	主屏分辨率、产品尺寸	100%	80%
W2	上市时间、手机外形、主屏参数、数据业务、摄像头	-	100%	100%
W3	产品类别、生产厂商、上市日期、手机类型、屏幕色彩、主屏尺寸、电子词典	-	100%	100%
W4	商家报价、手机类型、屏幕色彩、和弦铃声、摄像头	上市日期	100%	87.5%

4.4.2 实验分析

从表4-14可以看出，在W1标识的网站中，主屏分辨率、产品尺寸为两个未发现的属性，W4标识的网站中有未发现属性上市日期。

对W1标识的网站，从表4-10可以看出，在待抽取属性名与已抽取属性名拥有相同的父节点可信度确定步骤中，参考价格、手机制式两个属性的属性名所在叶子节点拥有最近的祖先节点为 ，它们到该节点的路径长度为3，主屏分辨率、产品尺寸两个属性的属性名所在叶子节点与参考价格、手机制式两个属性的属性名所在叶子节点拥有最近的祖先节点也为 ，但是它们到该节点的路径长度为2，根据本章4.3.5设定的规则，该证据的可信度为-1。在待抽取属性名与已抽取属性名格式可信度步骤中，主屏分辨率、产品尺寸的属性名格式记录为（”主屏分辨率”,””,”span”）、（”产品尺寸”,””,”span”），通过表4-9我们可以发现它们和参考价格、手机制式的属性名格式记录不相同，则该步骤中的可信度也为-1。由于主屏分辨率、产品尺寸属性名背景特征串为””，通过表4-8以及规则4.3.4，我们可以得出，该步骤可信度为0。虽然它们的字体关系可信度和待抽取属性名与用户感兴趣范围关系可信度都等于1，而它们的背景关系可信度都等于0，根据公式（4-6），主屏分辨率、产品尺寸需要抽取的可信度为0，由于本章设定的待抽取属性需要抽取可信度需要大于阀值Threshold，所以丢弃了这两个属性。

对于W4标识的网站，从表4-13我们可以看出，在字体关系可信度中，上市日期属性名字体特征串为（” h12”），通过表4-7我们发现它与参考价格、手机制式属性名字体特征串不相同，所以该步骤可信度为-1。同时在属性名格式特征可信度步骤中，上市日期属性名格式记录为（”上市日期”,””,”span”），通过与表4-7中对应的参考价格和手机制式属性名格式记录比较，根据规则4.3.6，该步骤可信度也为-1。而上市日期的属性名背景特征串为””，则根据表4-8和规则4.3.4得出该步骤可信度为0，所以虽然该属性在待抽取属性名与已抽取属性名拥有相同的父节点可信度以及用户感兴趣范围关系可信度确定步骤中都为1，根据公式（4-6），上市日期需要抽取的可信度为0，此时同W1网站主屏分辨率、产品尺寸属性一样，我们发现过程中丢弃了此属性。

但是总的来说，采用本章的方法，其准确率还是比较高，如在W2、W3网站中，本章的方法能发现全部需要抽取的属性，所以该方法在实际应用中具有较好的效果。

4.5 本章小结

本章提出了一种基于可信度的Web页面主题新属性发现方法，该方法通过引入可信度理论，并定义一系列的证据和规则，量化待抽取属性需要抽取的可信度，根据待抽取属性需要抽取的可信度发现Web页面中主题的新属性，并在部分网站中选取手机参数网页，通过实验对本方法进行验证，实验证明本章的方法能够较好的发现页面中主题的新属性。

第五章 垂直搜索引擎原型系统设计

本章通过综合本文提出的 Web 页面主题信息抽取方法和 Web 页面主题新属性发现方法，并参考现今搜索引擎的框架，设计了一种垂直搜索引擎原型系统。

5.1 系统整体框架

5.2 总体结构

根据第二章提出的垂直搜索引擎原理，本文设计的垂直搜索引擎原型系统框架如 5-1 图所示：

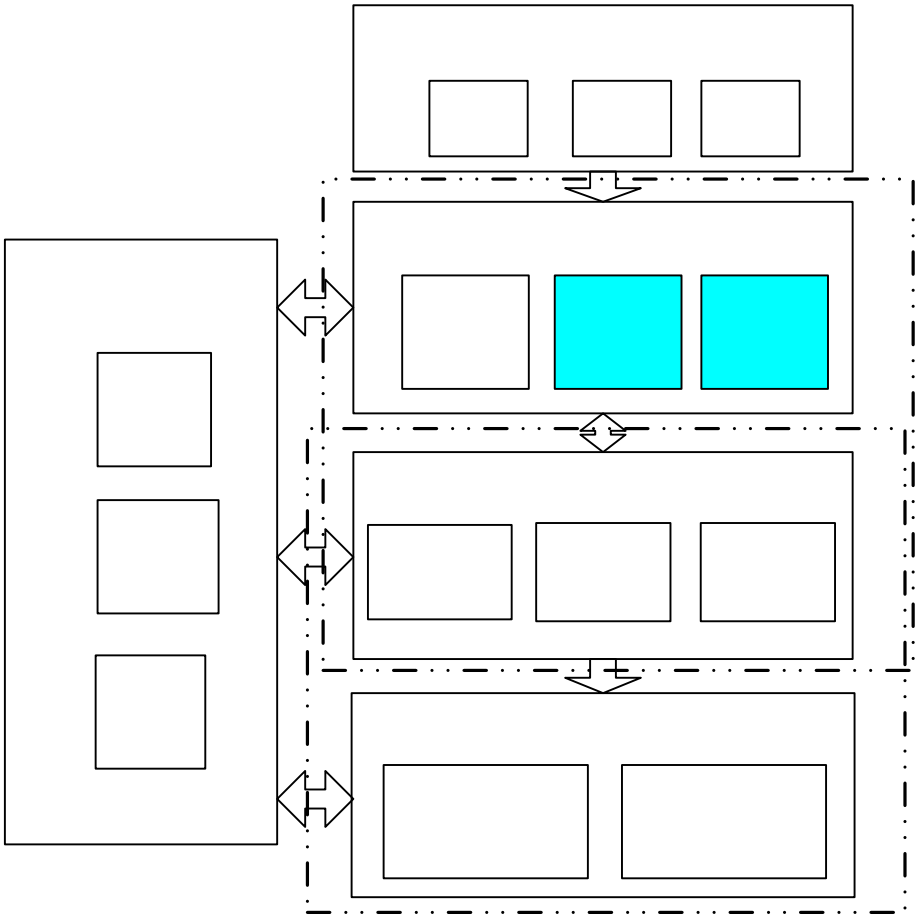

图 5-1 垂直搜索引擎原型系统框架图

从图 5-1 可以看出整个系统主要包含专业网络蜘蛛、用户接口、信息索引模块和系统管理模块。

其中数据源主要包括三个方面的数据，一是从互联网上获取，二是从异地数

专业网

据库中直接获得信息，由于垂直搜索引擎提供服务方本身的站点可能有一定的数据，还可以从本地数据和本地文件系统中获得信息，而本文设计的原型系统选取的数据源主要专注于从互联网上获得的 Web 页面。

5.2.1 模块介绍

专业网络蜘蛛模块的职责从互联网上采集与主题相关的信息。本原型系统主要的创新是专业网络蜘蛛模块，在专业网络蜘蛛模块中首先对页面的链接地址进行预测和分类；再利用了本文提出的基于标签序列的 Web 页面主题信息抽取方法，抽取页面中用户感兴趣的相关主题信息；当不能在页面抽取主题信息时，此时利用本文基于可信度的 Web 页面主题新属性发现方法判断待抽取页面中是否有主题的新属性，当出现新属性时，通过样本训练，生成新的页面抽取规则，从而实现新结构页面主题信息抽取的功能。

信息索引模块包括索引信息库建立和索引信息库维护两个子模块。索引信息库建立子模块先对采集的文档等信息进行分析，以决定如何对采集的文档等信息建立索引，然后将信息存入索引信息库；索引库的维护子模块是由于随着索引信息库中新索引信息的增加，需要对原来库中已存在信息进行相应的维护，同时还需要根据用户的查询请求，将用户感兴趣的信息以特定的形式返回给用户接口模块。

用户接口模块是用户与垂直搜索引擎原型系统的接口，它的作用是提供用户与搜索引擎交互操作的界面，以使用户更方便、更有效的使用搜索引擎查找信息；同时该模块还需要对用户提出的查询条件进行分析，以确定用户感兴趣信息范围，并将分析后的条件提交给信息索引模块，由信息索引模块根据条件将结果返回给用户，再以特定的显示风格将结果呈现给用户。

系统管理模块是在系统运行过程中对其他模块的维护和管理，它包含三个子模块，分别是系统初始化、出错信息管理以及日志管理。在系统的运行过程中，特别是专业网络蜘蛛模块，需要对一些模块的运行过程设定一些初始值，这需要由系统管理模块控制；同时在每个模块的运行过程中都可能出现错误信息，此时需要记录这些错误信息，以便开发人员定位并修改这些错误，同时对部分错误信息还需要系统自动调控，以使得用户获得满意的结果；日志管理是对系统运行过程中的日志进行记录，以方便用户分析整个系统的性能，并能根据日志找出部分系统错误出现的原因。

5.2.2 系统处理流程

从图 5-1 可以看出，垂直搜索引擎系统分为两个子系统，分别为信息采集子

系统和信息检索子系统。

信息采集子系统负责从互联网采集相应的主题信息，采集的信息进行分类等处理，存入索引信息库；信息检索子系统是根据用户的查询请求从索引信息库中返回最接近用户需要的信息。两个子系统相互独立的工作，但在功能上相互影响，信息检索子系统是对信息采集子系统功能的检验，信息采集子系统决定用户使用信息检索子系统的满意程度，以下分别对两个子系统的处理流程进行介绍。

1）信息采集子系统

信息采集子系统主要包含专业网络蜘蛛模块和信息索引模块中的索引信息库建立与维护子模块，图 5-2 描述了信息采集子系统的原理图。

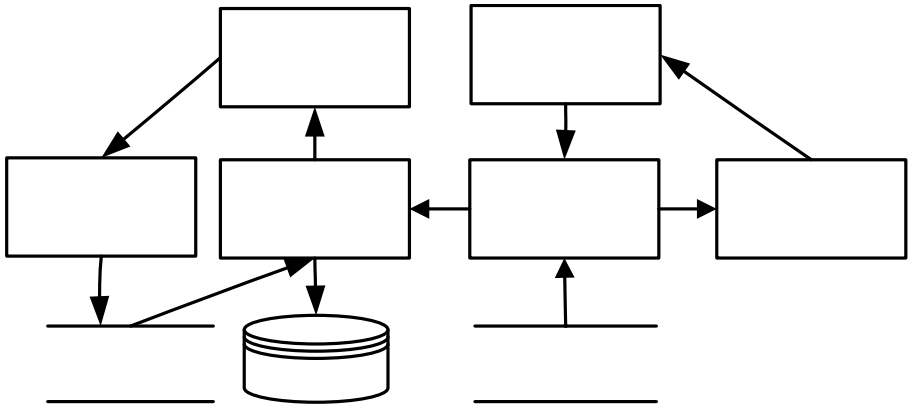

图 5-2 信息采集子系统原理图

它的工作流程如下：

1、通过初始化 URL 队列，选择一个链接，并下载该链接所示页面的源代码。

2、对页面源代码进行分析，提取出页面中链接，并通过 URL 队列管理模块，维护页面中提取的链接。

3、对页面源代码进行分析，根据规则库中的规则判断可否抽取页面中的主题信息，如果可以抽取页面中主题信息，转 4；否则转 5。

新属性发现

4、根据规则中的位置向量和主题属性格式文件抽取页面中主题信息，并为抽取出的主题信息建立索引，将其存入索引信息库中，同时维护原来索引信息库中的信息，转下一个页面处理。

5、根据本文提出的基于可信度的 Web 页面主题新属性发现方法判断页面中是否有主题新属性，如果有则转 6；否则转下一个页面处理。

样本训练 主题信息抽取

6、通过样本训练生成新的页面抽取规则，转 7。

7、将新生成的规则存入规则库中，转下一个页面处理。

2）信息检索子系统

信息检索子系统主要包含用户接口模块和信息索引模块中查询结果返回子模块，图 5-3 是信息检索子系统的时序图。根据该时序图，信息检索子系统的主

规则库 索引信息库

要工作流程如下：

1、用户通过搜索引擎提供的图形界面提交查询关键字。

2、通过查询条件分析器对用户提交的关键字进行分析，确定用户感兴趣信息的最大范围，如果提交的关键字为中文还需要对用户提交的关键字进行分词。

3、将由查询条件分析器产生的用户感兴趣信息的范围提交给信息索引模块，根据范围返回需要的查询结果。

4、通过查询结果分析器对查询结果进行处理，以特定的风格将查询结果呈现给用户。

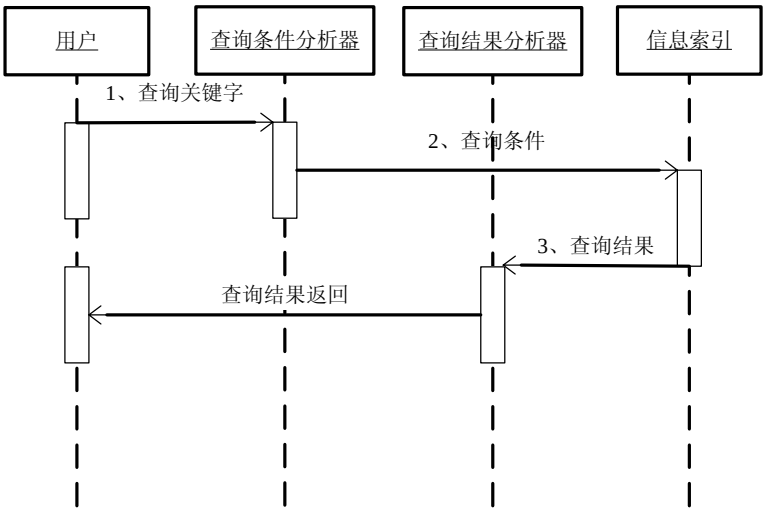

图 5-3 信息检索子系统时序图

在系统的运行过程中，系统管理模块提供了对整个系统的维护功能，它涉及到各个步骤出错信息和日志管理。

在上述两个子系统中，信息检索子系统可以通过开源软件 Lucene 实现它的功能，信息采集子系统中索引信息库建立和维护子模块也可以通过 Lucene 提供的 API 来实现。所以本系统主要是专业网络蜘蛛模块的设计，随后本章将对该模块进行详细设计和说明。

5.3 专业网络蜘蛛

5.4 总体设计

专业网络蜘蛛是信息采集子系统的关键模块，包含 URL 队列管理、主题信息抽取和新属性发现三个子模块，图 5-4 描述了专业网络蜘蛛模块的类图。

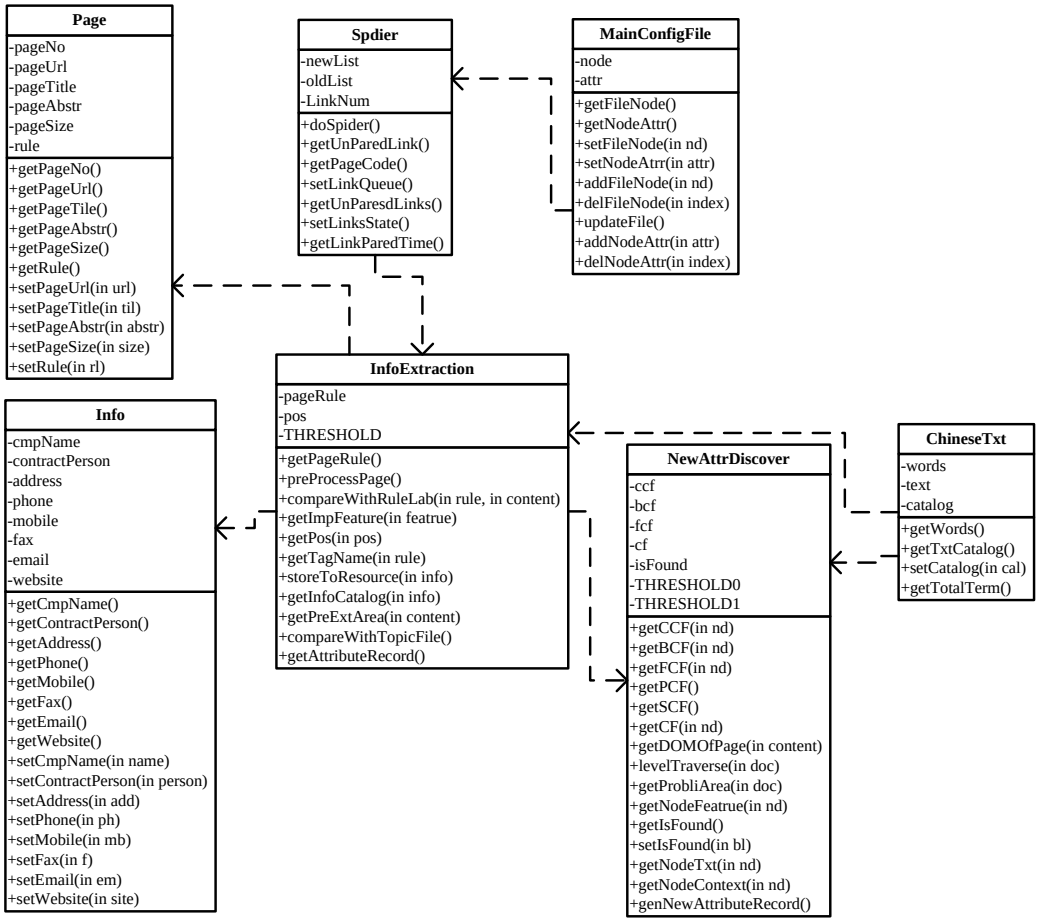

图 5-4 专业网络蜘蛛类图

专业网络蜘蛛模块主要有 URLQueneManagerImpl、Spider、PageInfoExImpl、NewAttributeDiscoverImpl 四个类，URLQueneManagerImpl 类主要实现 URL 队列管理模块的功能，Spider 是网络蜘蛛的主程序运行类，它实现了对整个网络蜘蛛运行过程的管理，PageInfoExImpl 类主要实现了页面主题信息的抽取，NewAttributeDiscoverImpl 类主要功能是判断页面是否出现主题的新属性。四个类主要方法说明如下。

URLQueneManagerImpl 类主要方法说明：

getLinkLevel：根据页面中的标签、链接的锚文本等预测链接的重要程度。

getLinkGroup：根据页面中的标签、链接的锚文本等预测链接的类别。

addToLinkQueue：将页面中新发现的链接存入 URL 队列中，以方便 Spider 类调用。

PageInfoExImpl 类主要方法说明：

storeToResourcesLib：提供与信息索引模块的接口，将抽取出的信息存入索引信息库中。

cmpWithRuleLib：将页面标签序列与规则库中规则的主题区域标签序列进行对比，生成待抽取区域集合。

compareWithTopicFile：根据页面的内容向量、规则中位置向量和主题属性格式文件判断是否可以抽取页面的信息。

NewAttributeDiscover 类主要方法说明：

levelTraverse：层次遍历 DOM 树，以生成待抽取属性需要抽取的可信度。

Spider 类主要方法说明：

doSpider：网络蜘蛛主程序。

5.4.1 URL 队列管理子模块

URL 队列管理子模块主要是对从页面中抽取出的链接进行管理，它是网络蜘蛛的重要组成部分。为了提高页面的覆盖率，通常网络蜘蛛需要寻找尽可能多的链接进行遍历，虽然对于垂直搜索引擎需要获取的页面数量比通用搜索引擎减少了很多，但是依然需要对一些不是很重要的网页进行处理，因为在网络蜘蛛抓取页面的过程中是将整个互联网看成一个图进行处理，图中各个节点代表互联网上的页面，而链接代表了图中各个节点之间的边，当某一链接虽然与主题信息不相关，但是如果它是连接另一部分相关主题页面的重要边时，此时的该链接将变的非常重要，所以依然需要对它进行索引处理。这样虽然垂直搜索引擎在一定程度上减少了链接的数量，但是还是需要对网络蜘蛛的抓取策略进行处理。

其具体功能如下：

1、URL 去重检测。由于 Web 上各个页面包含的大量重复的链接，如果不对链接去重，在很大程度上将浪费网络蜘蛛运行的效率，因此 URL 去重检测是 URL 队列管理的重要功能。

2、URL 队列分组和排序。根据各个链接的重要程度的不同，需要对待抽取的链接进行分组和排序，以决定链接抽取的先后顺序。

3、标签等关键页面信息的获取。主要根据页面中 Meta 和页面标题以及链接的锚文本(anchor text)判断链接的重要程度；

根据 URL 队列管理模块的功能，图 5-5 描述了 URL 队列管理模块的处理流程，其处理流程如下：

1、初始化 URL 队列集合。

2、下载页面，并分析页面源代码，抽取页面中链接。

3、去掉重复抽取的链接。

4、分析抽取的链接，确定链接的分组。

5、将分组后的链接进行排序，并存入 URL 队列中。

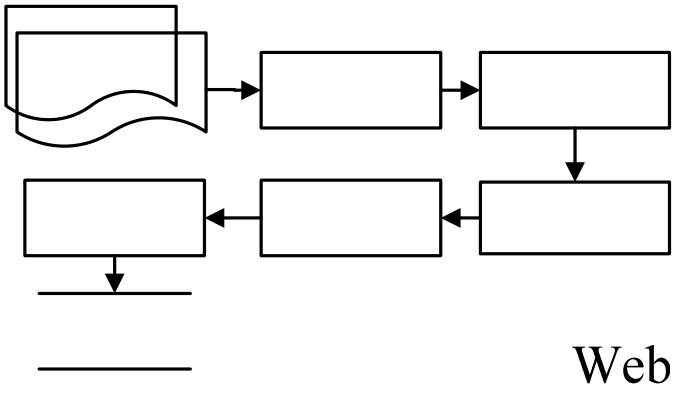

图 5-5 URL 队列管理模块的处理流程

5.4.2 主题信息抽取与新属性发现子模块

主题信息抽取模块主要根据本文第三章提出的基于标签序列的 Web 页面主题信息抽取方法抽取出页面中的主题信息。在主题信息抽取模块中，当不能利用抽取规则抽取 Web 页面中主题信息时，根据本文第四章提出的基于可信度的 Web 页面主题新属性发现方法判断页面中是否有新的主题属性，当出现新的主题属性时，通过样本训练，生成新的抽取规则，从而实现能抽取新结构页面的主题信息。

综合两个子模块，其具体的功能有：

1、页面主题信息抽取。根据规则库中的规则抽取页面的主题信息。

2、发现页面中主题的新属性。根据 DOM 树和已抽取的属性，生成待抽取属性需要抽取的可信度，以判断页面是否有新属性。

3、生成新的页面抽取规则。如果出现页面出现主题的新属性，则通过样本训练生成新的页面抽取规则。

主题信息抽取与新属性发现两个子模块相互协调工作，实现抽取页面中的主题信息，图 5-6 描述了这两个模块相互工作的工作流程。

其处理流程如下：

1、输入一个 Web 页面，对页面进行预处理，并生成页面标签序列、内容向量。并根据页面链接地址生成该页面所在网站的域名。

2、根据第一步生成的网站域名返回规则库中包含该域名所有规则，将返回规则中的主题区域标签序列和第一步中生成的页面标签序列对比，生成页面待抽取区域集合，当待抽取区域集合不为空时，转 3，否则转 6。

3、顺序选择待抽取区域集合中一个待抽取区域，并在集合中删除此区域。根据规则库中规则的位置向量以及主题属性格式文件，判断是否可以抽取该区域中标签的内容。当能抽取该区域中标签的内容，转 4，否则转 6。

4、根据规则库中规则的位置向量以及主题属性格式文件，抽取出该区域的主

题信息，并将抽取出的主题信息通过 Lucene 提供的接口将信息存入索引信息库中，转 5。

5、判断待抽取区域集合中是否为空，不为空则转 3，否则转下一个页面处理。

6、建立页面 DOM 树，根据 DOM 树生成待抽取属性需要抽取的可信度，根据待抽取属性需要抽取的可信度判断是否在页面中发现主题的新属性，如果没有发现主题的新属性，转下一个页面处理；否则，转 7。

7、对该页面通过样本训练生成位置向量、页面的标签序列、主题属性格式文件和网页所在网站的域名，归纳新的抽取规则，并将新的抽取规则存入规则库中，转下一个页面处理。

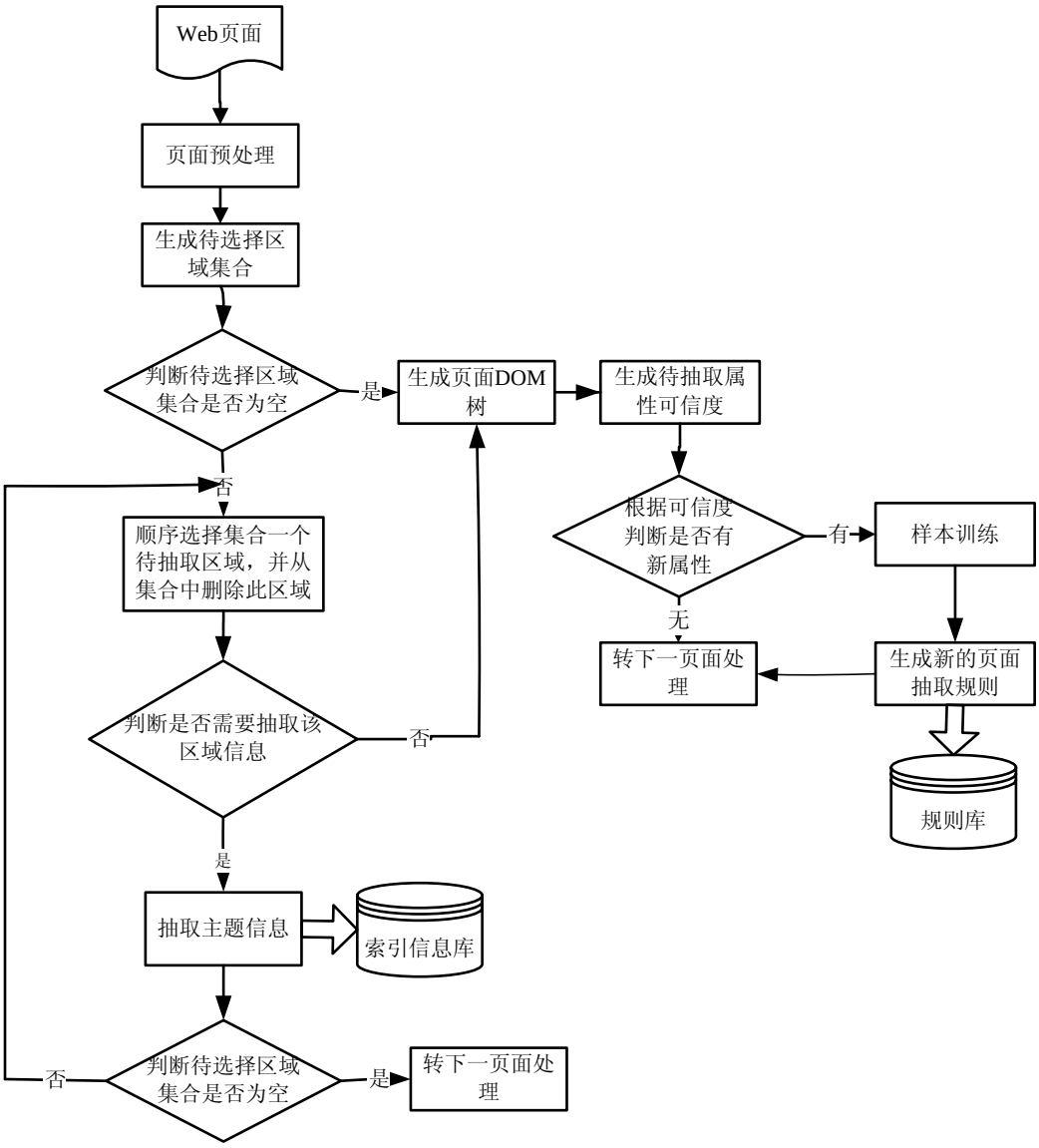

图 5-6 主题信息抽取与新属性工作流程图

5.5 本章小结

本章设计了一种垂直搜索引擎原型系统，该系统分为信息采集和信息检索两个子系统，本文对两个子系统中各个主要模块进行了详细的介绍，并根据搜索引擎的基本原理，介绍系统的处理流程。

由于本文研究的Web信息抽取方法主要是在专业网络蜘蛛模块中应用，本文主要对该模块进行了详细地设计。本文设计的专业网络蜘蛛模块包括三个子模块，分别是URL队列管理、主题信息抽取和新属性发现模块，三个模块相互协调工作，实现发现并抽取数据源信息。

第六章 总结和展望

6.1 本文工作总结

目前，网络在人们的生活中扮演着重要的角色。而随着互联网上信息的爆炸性增长以及更新速度快，日益增长的数据量极大困扰了人们获得感兴趣的信息，为更好地利用网络上丰富的信息资源，人们提出了不同的解决方法，其中垂直搜索引擎的诞生从一定程度上解决了这种问题。垂直搜索引擎在采集信息时需判定相关 Web 是否属于特定的主题，这就需要从 Web 页面等资源中抽取出特定的主题信息。无庸置疑，Web 页面信息抽取技术在垂直搜索引擎中占有极其重要的位置。因此，本文的主要工作也正是从 Web 页面中抽取与特定主题相关的信息。主要包括以下三点：

1、 针对现今 Web 信息抽取方法的不足：如需要构建巨大的知识库、页面处理过程复杂等，本文设计了一种基于标签序列的 Web 页面主题信息抽取方法，并建立该方法相应的模型，对该模型每个处理过程进行了详细的说明。通过实验将该方法应用到部分网站中页面的手机参数信息抽取中，实验证明应用本方法进行各个网站页面抽取时，其召回率和准确率都比较高，并且从一定程度上减少了 Web 页面处理时间，具有较好的实用性。

2、提出了一种基于可信度的 Web 页面主题新属性发现方法，该方法通过对待抽取属性自身结构和内容的分析，与已抽取属性的信息进行比较，并引入可信度理论，通过定义一系列规则和证据，量化待抽取属性需要抽取的可信度，再根据该可信度判断待抽取属性是否为主题的新属性，从而实现发现页面中主题的新属性。通过在实际网站中选取部分网页对本方法进行了验证，证明本方法能够较准确发现页面中主题的新属性。

3、结合第三、四章提出的方法，设计了一种垂直搜索引擎的原型，该原型主要创新是专业网络蜘蛛的设计，它综合了本文提出的基于标签序列的 Web 页面主题信息抽取方法和基于可信度的 Web 页面主题新属性发现方法，用于采集 Web 页面中的主题信息。并以 Lucene 开源软件建立和维护索引信息库，对用户的查询请求返回查询结果。

6.2 进一步展望

尽管本文在 Web 页面主题信息抽取及新属性发现方面做了一定的研究，但是仍存在一些问题有待进一步研究：

1、本文基于标签序列的 Web 页面主题信息抽取方法考虑的是手机名等文本信息的抽取，并未对页面中其他的信息如图片等多媒体信息进行处理，而多媒体信息在页面中的地位已经越来越重要，未来的工作需要在这一方面做出更进一步的努力。

2、在基于可信度的 Web 页面主题新属性发现方法中，本文定义了一系列证据和规则，通过这些规则和证据量化先验概率，具有较大的主观性，目前只是根据经验设定这些先验概率，而设定的这些先验概率对该方法的影响，本文暂未做更进一步研究，未来需要加强这方面的工作。

3、本文的仅完成了垂直搜索引擎原型系统的设计，未来的工作需要对原型系统进一步进行实现。

参考文献

[1] 中国互联网络信息中心.第 19 次中国互联网络发展状况统计报告. http://www.cnnic.cn/index/0E/00/11/ index.htm

[2] 中国互联网络信息中心.第 21 次中国互联网络发展状况统计报告. http://www.cnnic.cn/index/0E/00/11/ index.htm

[3] Robert.Techniques for Specialized Search Engines.In Proceedings of Internet Computing,2001.25~28

[4] Robert Baumgartner, Sergio Flesca, Georg Gottlob. Visual Web Information Extraction with Lixto.In:Proceedings of the 27th VLDB Conference, 2001.119~128

[5] Alberto H.F.Laender, Berthier A.RibeiroNeto, Altigran S.da Silva, Ju liana S. Teixeira. A Brief Survey of Web Data Extraction Tools. SIGMOD Record, 2002, 31(2):84~93

[6] D.Freitag. Machine Learning for Information Extraction in Informal Domains: [Ph.D].Carnegie Melton University,1998

[7] M.E.Califf.Relational Learning Techniques for Natural Language Information Extraction:[Ph.D.].University of Texas,1998

[8] S.Soderland.Learning Information Extraction Rules for Semistructured and Free Text.Machine Learning,1999,34(13):233-272

[9] William W.Cohen, Wei Fan.Learning Page-Independent Heuristics for Extracting Data from Web Pages. Proc 8th International World Wide Web Conference,1999

[10] YanHong Zhai, Bing Liu.Web Data Extraction Based on Partial Tree Alignment. In:Proc. 14th Int'l Conf. World Wide Web, 2005.76-85

[11] DC Reis, PB Golgher, AS Silva,et al.Automatic web news extraction using tree edit distance.In:Proceedings of the 13th International Conference on WWW2004, 2004. 504-505

[12] 李效东,顾毓清.基于 DOM 的 Web 信息提取.计算机学报.2002,25(5):78-85

[13] 陈琼,苏文键.基于网页结构树的 Web 信息抽取方法.计算机工程.2005, 31(20):54~55,140

[14] Peng Cai, Shipeng Yu, Ji-Rong Wen, et al.Extracting Content Structure for Web Pages Based on Visual Representation.In:5th Asian-Pacific Web Conference(ApWeb)Xian, 2003. 406~417

[15] Homepage of BYL data extraction research group. LRL:http://www. deg.byu.edu

[16] Ning Zhang,Hong Chen, Yu Wang,et al.Odaies: Ontology-driven Adaptive Web Information Extraction System.In:Proceedings of the IEEE/WIC International Conference on Intelligent Agent Technology.2003. 454~460

[17] 张成洪, 王向安, 古晓洪. 利用 Ontology 和规则表达式的 Web 信息抽取.计算机工程. 2004, 30(5):58~60

[18] 廖乐健, 曹元大, 李新颖. 基于 Ontology 的信息抽取. 计算机工程与应用, 2002.23:110~113

[19] Andrew McCallum,Dayne Freitag,Fernando Pereira.Maximum Entropy Markov Models for Information Extraction and Segmentation.In:17th International Conf.on Machine Learning, 2000.591~598

[20] Soumya Ray,Mark Craven. Representing Sentence Structure in Hidden Markov Models for Information Extraction.In:Proceedings of the 17th International Joint Conference on Artificial Intelligence, 2001.1273-1279

[21] 王胜,朱明. 基于最大熵马尔可夫模型的地址信息抽取. 计算机工程与应用.2005,21:192~194

[22] 钟敏娟, 郝谦, 刘云中.基于多模板隐马尔可夫模型的文本信息抽取算法. 计算机工程, 2006, 32(2):203~205

[23] Sergey Brin,Lawrence Page.The anatomy of a search engine.In:Proc. of the 7th International WWW Conference, 1998.14~18

[24] 郭谢.基于 Web Community 识别的专业搜索引擎研究: [硕士学位论文]. 杭州: 浙江大学,2006

[25] Andrew McCallum,Kamal Nigam, Janson Rennie, et al. Building Domain-Specific Search Engines with Machine Learning Techniques. In:Proc.AAAI-99 Spring Symposium on Intelligent Agents in Cyberspace, 1999

[26] Zaiqing Nie, Ji-Rong Wen, Wei-Ying Ma. Object-level Vertical Search.In:3rd Biennial Conference on Innovative Data Systems Research(CIDR),2007

[27] W3C.DOM. http://www.w3c.org/DOM/

[28] Lobobrowser.Cobra. http://www.lobobrowser.org/cobra.jsp

[29] Nicholas Kushmerick,Daniel S.Weld.Wrapper Induction for Information Extraction.In:Intl.Joint Conference on Artificial Intelligence,1997.246

[30] Arnaud Sahuguet, Fabien Azavant.WysiWyg Web Wrapper Factory(W4F). Technical report,University of Pennsylvania,Department of Computer and Information Science,1998

[31] J.Hammer,H.Garcia-Molina,J.Cho,et al.Extracting Semistructured Information from the Web.In:Proc. WorkShop Management of Semistructured Data,1997.136~144

[32] W.Han, D,Buttler, C.Pu.Wrapping Web Data into XML, SIGMOD Record, 2001,23(30):33~38

[33] David Buttler, Ling Liu, Calton Pu. A Fully Automated Object Extraction System for the World Wide Web.In:Proceedings of the 2001 International Conference on Distrubuted Computing Systems, 2001.361~370

[34] Nikolaos K.Papadakis, Dimitrios Skoutas. STAVIES: A System for Information Extraction for Unknown Web Data Sources through Automatic Web Wrapper Generation Using Clustering Techniques. IEEE Computer Society,2005,12(17):1638~1652

[35] Bing Liu,Robert Grossman,Yanhong Zhai.Mining Data Records in Web Pages. In:Proceedings of the 9th ACM SIGKDD international conference on Knowledge discovery and data mining,2003.601~606

[36] 李保利,陈玉忠,俞士汉.信息抽取研究综述.计算机土程与应用,2003,10:4~8

[37] Chinchor N, Marsh E. MUC-7 Information Extraction Task Definition,In Proceedings of the Seventh Message Understanding Conference, 1998

[38] David D.Lewis.Naive(Bayes) at Forty:The Independence Assumption in Information Retrieval.In:Proceedings of the 10th European Conference on Machine Learning,1998.4-15

[39] Yiming Yang. An evaluation of statistical approaches to text categorization. Information Retrieval Journal, 1999, 22(1):69~90

[40] Thorsten Joachims.Text Categorization with Support Vector Machines: Learning with Many Relevant Features.In:Proceedings of ECML-98,10th European Conference on Machine Learning,1998.137~142

[41] SebastianiF.Machine learning in automated text categorization.ACM Computing Surveys,2002,34(1):1~47

[42] Aitao Chen. Chinese Word Segmentation Using Minimal Linguistic Knowledge.In:Proceedings of the Second SIGHAN Workshop on Chinese Lanuage Processing,2003.148~151

[43] 张春霞,郝天永.汉语自动分词的研究现状与困难.系统仿真学报,2005,1(17):138~143,147

[44] Apache.Lucene.http://www.lucene.apache.org/

[45] Michael Chau, Hsinchun Chen. Comparision of Three Vertical Search Spiders. IEEE Computer Society,2003.56~62

[46] Jon M.Kleinbreg.Authoritive Sources in a Hyperlinked Enviroment.In:Proc.9th ACM-SIAM Symposium on Discrete Algorithms, 1999.604~632

[47] Benjamin Habegger, Mohamed Quafafou.Building web Information extraction tasks.In:Proccedings of the IEEE/WIC/ACM International Conference on Web Intelligence,2004.349~355

[48] Tak-Lam Wong, Wai Lam.A Probabilistic Approach for Adapting Information Extraction Wrappers and Discovering New Attributes.In:Proceedings of the Fourth IEEE International Conference on Data Mining,2004.257~264

[49] 王亮,朱征宇.基于扩展标记图的 Web 信息抽取器.计算机工程, 2005,31(8): 159~161,191

[50] William W.Cohen,Matthew Hurst,Lee S.Jensen.A flexible learning system for wrapping tables and lists in HTML documents.In:Proceedings of the 11th international conference on World Wide Web,2002.232~241

[51] Arnaud Sahuguet,Fabien Azavant.Building intelligent web applications using lightweight wrappers.International Journal of Data and Knowledge Engineering,2001,36 (3):283~316

[52] Davi De Castro Reis, Paulo B.Golgher, Alberto H.F.Laender,et al.Automatic web news extraction using tree edit distance.In:Proceedings of the 13th international conference on World Wide Web,2004.502~511

[53] 唐世渭,杨冬青,王腾蛟,等.基于 Ontology 的 Web 内容二阶段半自动提取方法. 计算机学报.2004,27(3):310~318

[54] Masahiro Tanaka,Toru Ishida..Ontology extraction from tables on the Web. In:Proceedings of the International Symposium on Applications on Internet,2006.284~290

[55] Harith Alani,Sanghee Kim,David E.Millard,et al.Automatic Ontology-Based Knowledge Extraction from Web Documents.IEEE Intelligent Systems,18(1):14~21

[56] Suhit Gupta,Gail Kaiser,David Neistadt,et al.DOM-based Content Extraction of HTML Documents.In:Proceedings of the 12th international conference on World Wide Web,2003.207~214

[57] Robert Baumgartner, Sergio Flesca, Georg Gottlob. Visual Web Information Extraction with Lixto.In:Proceedings of the 27th International Conference on

Very Large Data Bases,2001.119~128

[58] D.W.Embley,D.M.Campbell,Y.S.Jiang,et al. Conceptual-Model-Based Data Extraction from Multiple-Record Web Pages. Data&Knowledge Engineering,1999, 31(3):227~251

[59] Mihai Surdeanu,Sanda M.Harabagiu.Infrastructure for Open-Domain Information Extraction.In:Proceedings of the second international conference on Human Language Technology Research,2002.325~330

[60] 蔡自兴,徐光佑.人工智能及其应用.北京:清华大学出版社,2005.102~107

致 谢

本文得以顺利完成，首先要向我的导师李建华教授致以崇高的敬意和深切的谢意。衷心感谢李老师多年来的支持、鼓励、关怀和信任，本文是在李老师的精心指导下完成的。自从我进入实验室以来，李老师始终以他宽阔的视野、渊博的学术知识和严谨的治学态度对我言传身教，并将使我受益终生。

感谢实验室李桂林、李勇军、曾慧琼、张慧、李鹏等同学对我生活上的关心和学习上的指导，你们的帮助，让我不仅在学术上获益良多，开阔了视野，同时也进一步提高了自己的能力和团队合作的优良品质，在三年的研究生生涯中，我们一起留下了一段美好的回忆。

同时，感谢父母为我成长付出的心血，感谢他们对我多年来的对我的培养和教育，在我成长道路上，他们一直鼓励和支持我。

再次向所有帮助我的人致以诚挚的感谢。

攻读学位期间主要的研究成果

已发表的学术论文：

1、一种自动抽取 Web 信息方法的设计与实现. 胡国晴, 李建华. 计算机与现代化. 已录用.

2、一种基于可信度分析的 Web 页面新属性发现方法. 胡国晴, 李建华. 计算机技术与发展. 已录用.

参加的科研项目：

1、参与湖南科创信息股份有限公司内容管理系统开发.

2、参与湖南省信息统计调查系统开发.

3、参与常德卷烟厂 OA 系统开发.