中文摘要

现今通用搜索引擎仅能收录 Web 上通过链接可以爬行到的页面部分。然而对于大量的深度网资源，由于搜索引擎的爬虫无法通过链接爬行到这些页面，因而搜索引擎无法索引到这部分信息。据统计，目前深度网资源量是普通可索引到的资源的 500 倍左右。这些信息隐藏在 Web 页面的查询表单（深度网入口）后面，保存在大型的动态数据库中。如此庞大的信息资源如果没有合理的、高效的方法去获取，无疑将是巨大的损失。此外，深度网的研究涉及数据集成、中文语义识别等多个领域。因此，对于深度网爬行技术的研究具有极为重大的现实意义和理论价值。

目前的研究表明，深度网资源涉及的领域广泛且深度网入口形式缺乏统一规格，因而深度网资源不可能做到统一的集成，只能针对某一领域进行研究。基于此，本文设计并实现了一种结合深度网爬行技术在内的主题领域爬虫系统。系统旨在对某一领域进行包括深度网资源在内的全方位的爬行，以获得更全面、更优质的主题资源。系统采用基于本体域的入口定位及基于网页标签距离及语义判别的方法抽取入口模式。并且对主题特征词的学习采用一种在线学习的特征词训练方法。实验表明，爬虫可以较好地发现深度网资源，实现了对包含深度网信息在内的主题资源的大量获取，获得更多更丰富的信息。

关键字　主题爬虫，搜索策略，深度网，入口模式(schema)，本体域

Abstract

Nowadays general search engines can only index pages that can be crawled through link. However, as to the great amount of deep web resources, search engines can not index them because crawlers never reach these pages. According to statistics, the total amount of deep web resources is about 500 times that of web which can be crawled by crawlers. This information is hidden behind the query forms of web pages (deep web interface) and is stored in large dynamic databases. There is no doubt a huge loss if no rational and efficient ways can be used to obtain these resources. In addition, deep web researches involve data integration, Chinese language semantic recognition and so on. For this reason, the research of deep web crawling technology has extremely important practical significance and theoretical value.

Currently researches show that it is impossible to integrate all domains of deep web resources because of broad-areas of them and lack of uniform interface schemas. Based on this, we designed and implemented a theme domain crawler containing crawling the deep web resources. The system aims to crawl all resources containing deep web information in the domain and to obtain more comprehensive and high quality of the theme resources. A method of deep web interface location based on ontology and another one of interface schemas extraction based on the distances between webpage tags and semantic recognition are adopted in the crawler. In addition, a method of theme features online learning is also used in it. Experiments indicate that the crawler can discover deep web resources effectively and obtain a great amount of the theme resources and get much more abundant and richer information.

Keyword Focused Crawler, Search Strategy, Deep Web, Interface Schema, Ontology

图 目 录

图 2.1 通用爬虫模型 7

图 2.2 主题爬虫模型 8

图 2.3 VSM 模型示意图 11

图 3.1 深度网资源的用户访问过程[21] 15

图 3.2 深度网爬虫的爬行过程[21] 15

图 3.3 FFC 体系结构[1] 16

图 3.4 ACHE 架构[2] 17

图 3.5 HIFI 18

图 3.6 匹配两个关系模式：Personnel 和 Employee-Department ... 20

图 3.7 由关系模式 S1 获得的 DLG 图 G1 20

图 3.8 初始 Mapping(前 10 行) 21

图 3.9 过滤后的结果 21

图 3.10 相关度判别 22

图 3.11 药品查询表单(drug.39.net) 28

图 3.12 手机查询表单(mobile.sina.com.cn) 28

图 3.13 表单查询入口 31

图 3.14 同义词词林语义分类树状图[36] 33

图 3.15 数据文件格式举例 36

图 3.16 索引文件格式举例 36

图 3.17 查询入口表单的区域划分 39

图 4.1 整体框架图 42

图 4.2 系统整体流程图 45

图 4.3 主题爬虫程序主界面 46

图 4.4 URLlinkList 表字段 47

图 4.5 多线程爬虫的类定义 48

图 4.6 结点存储结构 49

图 4.7 各标签元素的结点结构 50

图 4.8 Distance、FormElemType、Attribute 类定义 51

图 4.9 深度网入口处理流程 53

图 4.10 Text、Document、Word 类的定义 54

图 4.11 九州通医药网爬行链接图 57

图 4.12 中国医药网爬行链接图 57

表 目 录

表 3.1 药品查询表单的本体域定义 29

表 3.2 哈工大扩展版编码规则表 34

表 3.3 《扩展版》的文件格式说明 35

表 4.1 比较结果 56

南开大学学位论文版权使用授权书

本人完全了解南开大学关于收集、保存、使用学位论文的规定，同意如下各项内容：按照学校要求提交学位论文的印刷本和电子版本；学校有权保存学位论文的印刷本和电子版，并采用影印、缩印、扫描、数字化或其它手段保存论文；学校有权提供目录检索以及提供本学位论文全文或者部分的阅览服务；学校有权按有关规定向国家有关部门或者机构送交论文的复印件和电子版；在不以赢利为目的的前提下，学校可以适当复制论文的部分或全部内容用于学术活动。

学位论文作者签名：

2008 年 5 月 30 日

经指导教师同意，本学位论文属于保密，在　　年解密后适用本授权书。

指导教师签名：		**学位论文作者签名：**	
解　密　时　间：	年　　月　　日		

各密级的最长保密年限及书写格式规定如下：

内部　5 年（最长 5 年，可少于 5 年）
秘密★10 年（最长 10 年，可少于 10 年）
机密★20 年（最长 20 年，可少于 20 年）

南开大学学位论文原创性声明

本人郑重声明：所呈交的学位论文，是本人在导师指导下，进行研究工作所取得的成果。除文中已经注明引用的内容外，本学位论文的研究成果不包含任何他人创作的、已公开发表或者没有公开发表的作品的内容。对本论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确方式标明。本学位论文原创性声明的法律责任由本人承担。

学位论文作者签名：陈名

2008年5月30日

第一章 导论

目前通用搜索引擎仅能收录 Web 上通过链接可以爬行到的页面部分。然而对于大量的深度网资源，由于搜索引擎的爬虫无法通过链接爬行到这些页面，因而搜索引擎无法索引到这部分信息。据估计，目前深度网资源量是普通可索引到的资源的 500 倍左右。这些信息隐藏在 Web 页面的查询表单（深度网入口）后面，保存在大型的动态数据库中。如此庞大的信息资源如果没有合理的、高效的方法去获取，无疑将是巨大的损失。因此，对于深度网爬行技术的研究具有极为重大的现实意义。

现今人们对深度网已有了一些研究，这些研究主要集中在对深度网资源的集成应用上。目前的研究表明，深度网资源涉及的领域广泛、门类齐全并且深度网入口形式缺乏统一规格，因而对 Web 上的所有的深度网资源不可能做到统一的集成，只能针对某一领域进行研究。基于此，本文设计并实现了一种结合深度网爬行技术在内的主题领域爬虫系统。系统旨在对某一领域进行包括深度网资源在内的全方位的爬行，以获得更全面、更优质的主题资源。

第一节 论文背景的介绍与问题提出

搜索引擎领域涉及的研究内容多种多样，对深度网的研究是这一领域的又一热点问题。本节将对深度网研究的背景和课题工作内容分别做简单介绍。

1.1.1 论文背景介绍

Web 信息的大量增长对于人们在网络中定位所需的信息提出了巨大的挑战。搜索引擎作为一种有效的工具，可以帮助人们在浩如烟海的网络信息中获取所需内容。然而通用的搜索引擎由于其索引的信息量大、涉及领域宽泛，通常一次查询返回大量结果而且这些结果往往会包含各个领域的内容，导致大量无用信息的出现。人们往往会迷失在搜索返回的大量信息中，却无法找到自己想要的内容。垂直搜索引擎的出现在一定程度上解决了这一问题。垂直搜索引擎，即专业或主题搜索引擎，就是专为查询某一领域或主题的信息而产生的查询工

具，它专门收录某一主题的信息，对解决该领域内的实际查询问题要比通用搜索引擎有效得多。垂直搜索引擎是相对通用搜索引擎的信息量大、查询不准确、深度不够等问题提出来的新的搜索引擎服务模式，通过针对某一特定领域、某一特定人群或某一特定需求提供的有一定价值的信息和相关服务。其特点就是“专、精、深”，且具有行业色彩，相比较通用搜索引擎的海量信息无序化，垂直搜索引擎则显得更加专注、具体和深入。由于垂直搜索引擎只索引某一特定领域的页面信息，因此对该领域的垂直搜索引擎的查询不会返回其他领域的信息。这在一定程度上满足了用户的需要。与通用搜索引擎爬虫不同，垂直搜索引擎所使用的爬虫不爬行 Web 上的所有网页。它们只关心与本领域相关的页面，因此又称为主题爬虫或聚集爬虫。主题爬虫力求在尽可能少地遍历 Web 页面的前提下，尽可能多地发现主题相关的页面。因此，主题爬虫通常使用“Best First”策略爬行主题相关度最高的页面。对页面的主题相关度的计算通常依赖于主题特征词的选取，所以一个专业搜索引擎的爬虫的主题特征词选择的是否适合在很大程度上影响了它所爬行的页面的范围和内容。

令人遗憾的是，现今的搜索引擎并不能检索到 Web 上的每一处信息，有一部分动态网页是普通搜索引擎爬虫无法爬行到的。这一部分网页称为深度网(Deep Web)，与之对应的普通搜索引擎爬虫可以爬行的资源称为浅层网(Surface Web)。据统计目前这部分动态信息是根据页面链接所能爬行到的信息的 500 倍左右。更为可喜的是，这些信息通常为高质量的结构化数据，因此有效地获取深度网资源可以为用户提供更为高质量的信息来源。对这一领域的研究具有重要的理论价值和现实意义。深度网资源通常是指 Web 上的大型在线数据库。通过填写并提交某些查询性质的表单，服务器会根据查询条件从数据库中选择符合条件的信息记录并动态生成页面返回给用户。因此，对深度网资源的获取，通俗的讲，就是从 Web 上找到这样的深度网入口查询表单并识别其查询条件，然后模拟提交查询条件对网络数据库资源进行访问，以获得数据库信息。寻找这样的查询表单即对深度网入口的定位；识别表单查询条件即对深度网入口模式(schema)的抽取。因此对深度网入口的定位及其模式(schema)的抽取成为深度网爬行的关键问题。

1.1.2 问题提出

WWW 的规模持续以令人惊叹的速度增长。根据 Google 搜索引擎的索引库中

网页数量的统计，全球网页数量到 2005 年已经达到 80 亿。和网络规模的迅速膨胀形成鲜明的对比，尽管各个大型的通用搜索引擎都维护着庞大的索引，但是索引的规模增长远远不及网络本身。相对整个网络，他们仅能够覆盖一小部分。

通用搜索引擎力图覆盖整个网络，并为所有可能的主题提供查询服务。因此，这些通用搜索引擎收集的数据虽然信息量很大，但对于不同领域的用户的查询不能给出满意的结果。尤其是当搜索那些用户感兴趣的但通用搜索引擎认为页面重要度很低的页面时，费时费力，效果差。

面向主题的垂直搜索引擎克服了以上的缺点，拥有较好的查全率和查准率，因为它们将搜索网页的内容限定在一定的领域里，有效地划定了搜集的范围。一个面向主题的搜索引擎用一部分事先选定好的网页作为体现用户兴趣的样本。为了获得更多相关的网页，主题搜索引擎从一个给定的集合出发，递归地抓取集合中网页指向的所有相关于该主题的链接。对主题相关的网页进行索引，大大地提高了查询的准确率。

本文针对主题爬虫特征词的选取，提出一种对垂直搜索引擎主题爬虫特征词的训练方法。通过预先对一组样本进行训练，提取主题特征词并通过在线调整的方法对主题特征词进行优化以期获得更好的区别主题的能力。并且针对深度网爬行的问题提出一种基于本体域的深度网入口定位技术和基于网页标签距离及同义词比较的深度网入口模式抽取的方法。实现了对包含深度网信息在内的主题页面的全方位的爬行，以获得更多的有效信息。

第二节 综述与论文重点内容说明

目前在搜索引擎爬虫领域已有大量研究成果。本节首先对现今在搜索引擎、爬虫和深度网研究方面的成果做简单介绍，然后对本文针对的深度网做重点说明。

1.2.1 研究综述

目前在国内外继 Google、Yahoo!等传统的搜索引擎之后的新一代搜索引擎的研究正在成为一个热点，具有代表性的引擎如下。

1.Scirus 是面向科技文献的一个垂直搜索引擎，它的信息源主要包括网页和期刊两部分。它首先对网络中所搜索到的结果进行过滤，然后只列出包含有科学信息的成分，方便了科研人员的使用。

2.Berkeley 的 Focused Project 系统通过两个程序来指导爬行器，一个是分类器，用来计算下载文档与预定主题的相关度，另一个程序是净化器，用来确定那些指向很多相关资源的页面。

现今对深度网领域的研究成果有 L. Barbos 等人[1]提出的聚焦表单的深度网爬虫模型 FFC(Form-Focused Crawler)以及对其进行改进而成的适应性深度网表单入口的定位方法 ACHE(Adaptive Crawler for Hidden-Web Entries)[2]；K. C. -C. Chang 等人[3]开发的大规模深度网资源集成系统 MetaQuerier。

专业网络爬虫技术的研究成果有 Chakrabarti et al.实现的一个免定制和存储管理的专业爬行器[4]；Diligenti[5]提出的基于语境图的搜索策略，通过构建典型页面的 Web 语境图来估计离目标页面的距离；斯坦福大学的 Taher Haveliwala 提出的 Topic-Sensitive PageRank 算法[6]等待。

1.2.2 重点内容说明

由于深度网资源的快速增长，越来越多的焦点集中在获取深度网信息的技术上。一些技术和应用正在被尝试，以便更为方便地访问深度网信息，这包括：元搜索 MetaSearchers[7][8][9]，深度网爬虫[10][11]，在线数据库目录服务[12]和 Web 信息集成系统[3][13]。

目前，对于深度网已经有了大量的研究。这些研究主要集中在对深度网的入口定位和深度网入口的查询模式（schema）抽取两大部分。

对于深度网的入口定位技术主要应用在集成某一领域的深度网数据库信息，以备后续查询所用。这一应用中，最重要的一步工作就是对深度网入口进行有效地定位。然而对于在大量的网页中准确定位深度网数据库的入口点目前还是一项具有挑战性的任务。这主要是由于深度网资源在 Web 上分布的不均匀性或稀疏性所致。有实验表明，一个基于最优爬行策略的主题聚集爬虫爬行 100，000 个与电影相关的页面后，仅检索到 94 个电影查询表单。由于 Web 上的资源总是处于一种动态的平衡，即不断有新的资源加入，同时不断有旧的资源被删除或更新，因此自动地发现深度网数据库入口是很重要的。因此，为了能够有

效地维护一组实时更新的深度网资源的集合，必须采用一种有效的广度搜索策略，同时要避免陷入 Web 上大量的无深度网入口区域。

对于深度网入口查询模式的抽取是深度网研究中的基础性工作。其中的关键问题是对入口表单抽取出查询模式，即获得一组与表单内各标签元素对应的属性。在对标签元素与属性的匹配过程中，通常还会分为 1：1 和 1：m 的匹配，少数表单还会出现 m：n 的匹配。在 1：1 的匹配中，通常一个标签元素对应一条属性，例如一个文本框标签对应名称属性；在 1：m 的匹配中，通常多个标签元素对应一条属性，例如由分别代表年、月、日的三个下拉框标签对应一条日期属性。属性的匹配一般遵循对网页定义的若干规则，如标签间的距离等。这些规则通常通过对大量网页进行分析而提取出来的。此外，对某些表单的分析还设计到客户端脚本的解析。

深度网爬虫若想获得高质量的结果，判断表单的领域归属至关重要，有时甚至是必须的。例如，在深度网集成技术中，如果存在大量噪声，或包含不属于集成领域的表单，则集成结果的有效性将大打折扣。然而，自动化地爬行过程总是会检索不同领域的表单。聚焦主题的爬虫在爬行过程中总会遇到一种情况，即在某些主题网页中包含不同领域的查询表单。例如在包含机票查询的表单中有时也会包含酒店查询的表单。

此外，页面通常还会包含一些非查询性质的表单，比如登录表单、邮件订阅、quote request 和电子邮件表单。对此类表单的发现及排除也会对查询结果的正确性起到很大作用。

第三节 本文内容组织

论文的组织结构如下：

第一章，导论。介绍现今搜索引擎爬虫和深度网研究的现状和课题研究的意义以及课题的工作内容。

第二章，主题爬虫模型。介绍目前流行的通用爬虫及主题爬虫的一般模型结构及已有的相关技术。

第三章，系统关键技术。介绍现今的深度网入口的定位和解析的相关技术，并提出对本文所设计的爬虫的关键技术问题的解决方案。

第四章，实现与试验结果。介绍整个系统的结构设计思想和实现方法以及实验结果。

第五章，总结与展望。

第二章 主题爬虫模型

根据目前的研究，搜索引擎按其逻辑功能的不同，可分为五个模块：爬行器、分析器、索引器、检索器、用户接口。其中爬行器是一个搜索引擎的开始模块，也称网络爬虫、Spider、Crawler、Robot 等。其主要功能是根据一定的爬行策略，及时准确地获取 Web 上的网页信息并按时间段更新网页信息，避免死链接。

在组成搜索引擎的五个模块中，爬行器位于引擎的第一步，具有关键地位，同时也是区别通用搜索引擎和垂直搜索引擎的最主要的标准之一。本章将对现有的主题爬虫的模型及所使用的相关技术进行介绍。

第一节 通用爬虫模型

根据目前所使用的成熟的技术，通用爬虫模型从功能上主要分为页面采集器、页面解析器和链接过滤器三个模块，如图 2.1 所示。此外还有专门用于存放下载后的页面内容的页面库和存储待爬行的 URL 链接的 URL 库。

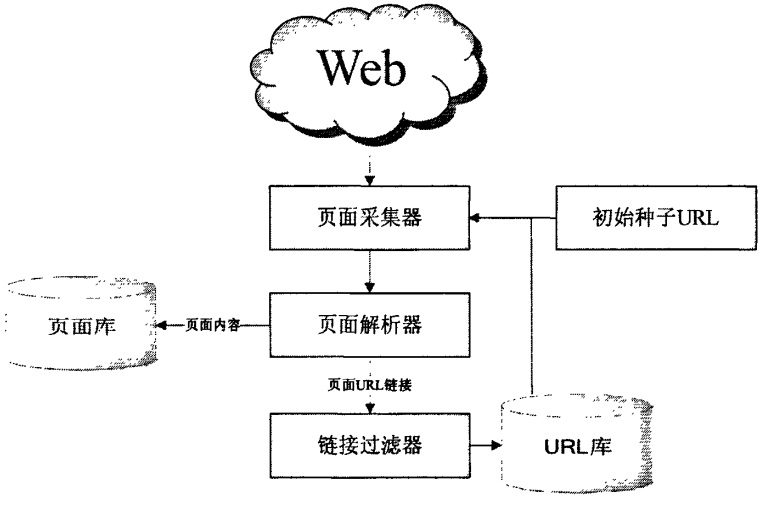

图 2.1 通用爬虫模型

通用爬虫的工作模式：页面采集器根据事先给定的种子站点 URL 进行爬行。

爬行到的页面交给页面解析器进行处理。页面解析器的主要功能是解析页面内容，提取链接并将页面信息存入页面库。提取的链接由链接过滤器处理。链接过滤器将格式正确的链接存入 URL 库，此外，还要进行 URL 的去重处理。存入 URL 库的链接将再次等待页面采集器来提取。

第二节 主题爬虫模型

图 2.2 描述了一个简单的主题爬虫的大体框架(不同设计思想的主题爬虫结构会有所不同)。从图中可以看出，主题爬虫通过主题判别器对页面相关度判断，以确定是否相关，将不相关的网页丢弃。对页面解析出的链接进行相关度预测，链接的预测相关度值大于某一阈值时，将链接存入 URL 库。

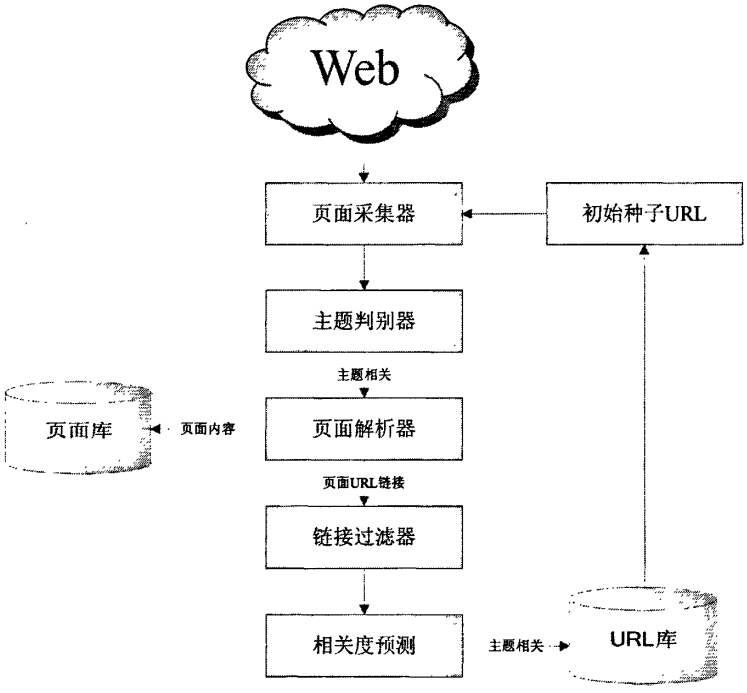

图 2.2 主题爬虫模型

主题爬虫与通用爬虫研究关注的焦点有所不同，其关键在于如何尽可能多地下载相关网页，避免与主题无关或低质量的网页，提高主题资源的覆盖度。

要达成这个目标显然依赖于能否在下载页面前，准确地预测页面与给定主题的相关度及其重要性；其次，选择好的种子 URL 集合、搜索策略以及主题表达方式也相当重要。

第三节 主题爬虫的关键技术

本节将介绍目前主题爬虫中常用的一些关键技术，包括分词、相关度计算和常用的爬行策略。以下分别介绍。

2.3.1 分词技术

中文分词是搜索引擎中的重要组成部分和关键技术。对于通用搜索引擎，分词技术主要应用于建立索引；对于垂直搜索引擎，除建立索引外，还应用在爬虫阶段。中文分词技术有基于字符串匹配的分词方法、基于统计的分词方法和基于理解的分词方法三种基本类型。

1．基于字符串匹配的分词方法

这种方法又叫做机械分词方法，它是按照一定的策略将待分析的汉字串与一个“充分大的”机器词典中的词条进行匹配，若在词典中找到某个字符串，则匹配成功（识别出一个词）。按照扫描方向的不同，串匹配分词方法可以分为正向匹配和逆向匹配；按照不同长度优先匹配的情况，可以分为最大匹配和最小匹配；按照是否与词性标注过程相结合，又可以分为单纯分词方法和分词与标注相结合的一体化方法。常用的几种机械分词方法如：正向最大匹配法（由左到右的方向）；逆向最大匹配法（由右到左的方向）；最少切分（使每一句中切出的词数最小）。

根据对每种算法的统计结果，单纯使用正向最大匹配的错误率为 1/169，单纯使用逆向最大匹配的错误率为 1/245。但这种精度还远远不能满足实际的需要。实际使用的分词系统，都是把机械分词作为一种初分手段，还需通过利用各种其它的语言信息来进一步提高切分的准确率。

2．基于理解的分词方法

这种分词方法是通过让计算机模拟人对句子的理解，达到识别词的效果。其基本思想就是在分词的同时进行句法、语义分析，利用句法信息和语义信息

来处理歧义现象。它通常包括三个部分：分词子系统、句法语义子系统、总控部分。在总控部分的协调下，分词子系统可以获得有关词、句子等的句法和语义信息来对分词歧义进行判断，即它模拟了人对句子的理解过程。这种分词方法需要使用大量的语言知识和信息。由于汉语语言知识的笼统、复杂性，难以将各种语言信息组织成机器可直接读取的形式，因此目前基于理解的分词系统还处在试验阶段。

3．基于统计的分词方法

从形式上看，词是稳定的字的组合，因此在上下文中，相邻的字同时出现的次数越多，就越有可能构成一个词。因此字与字相邻共现的频率或概率能够较好的反映成词的可信度。可以对语料中相邻共现的各个字的组合的频度进行统计，计算它们的互现信息。定义两个字的互现信息，计算两个汉字 X、Y 的相邻共现概率。互现信息体现了汉字之间结合关系的紧密程度。当紧密程度高于某一个阈值时，便可认为此字组可能构成了一个词。这种方法只需对语料中的字组频度进行统计，不需要切分词典，因而又叫做无词典分词法或统计取词方法。由于这种方法经常会抽出一些共现频率高但并非词的字组，如“这一”、“之一”等，因此，实际使用时要使用一部基本的分词词典（常用词词典）进行串匹配分词，同时使用统计方法识别一些新的词，即将串频统计和串匹配结合起来，既发挥匹配分词切分速度快、效率高的特点，又利用了无词典分词结合上下文识别生词、自动消除歧义的优点。

2.3.2 相关度判别技术

在传统的计算文档相似度的算法中，以 Salton 教授提出的向量空间模型(Vector Space Model)[14]应用最为广泛。向量空间模型基于这样一个关键假设，即组成文章的词条所出现的顺序是无关紧要的，它们对于文章的主题所起的作用是相互独立的，因此可以把文档看作一系列无序词条的集合。

VSM 模型以特征项作为文档表示的坐标，以向量的形式把文档表示成多维空间中的一个点，特征项可以选择字、词和词组等(根据实验结果，普遍认为选取词作为特征项要优于字和词组)表示向量中的各个分量。

它的基本思想是这样的，把文档 d_i 看成是由一组词条 $\{T_1,T_2,...T_n\}$ 构成的，对于每一个词条 T_i，都可以根据它在文档中的重要程度赋予一定的权值 w_i。可

以将$T_1,T_2,...T_n$看成一个 n 维坐标系，$w_1,w_2,...w_n$为对应的坐标值，因此每一篇文档都可被看作向量空间中由一组词条矢量构成的一个点，如图 2.3 所示。

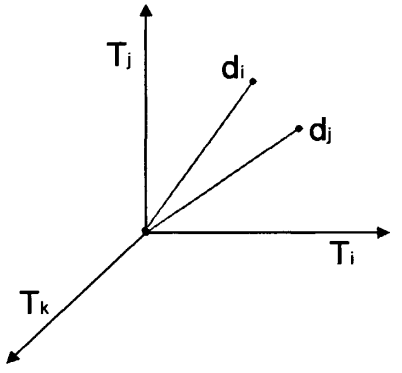

图 2.3 VSM 模型示意图

向量空间模型中文档的特征表示方法描述如下：

在向量空间模型中，设D是一个包含m篇文档的文档集合

$$D=\{d_1,...d_i,...d_m\},i=1,2,...,m \quad (2.1)$$

集合中的每篇文档d_i都被表示成如下形式的向量：

$$d_i=\{w_{i1},...w_{ij},...w_{in}\},i=1,2,...,m;j=1,2,...,n \quad (2.2)$$

其中，w_{ij}表示第j个特征项T_j在文档d_i中的权值。

权值的计算有以下几种方法：

1. $w_{ij}=\begin{cases}1 & \text{第}j\text{个特征项属于文档}d_i\\ 0 & \text{第}j\text{个特征项不属于文档}d_i\end{cases}$

2. $w_{ij}=\begin{cases}t_{ij} & \text{第}j\text{个特征项在文档}d_i\text{中出现的次数}\\ 0 & \text{第}j\text{个特征项不属于文档}d_i\end{cases}$

3. TF-IDF 词频统计方法[15]。该方法基于这样一个假设：在真实语料中，出现频率较高的词条(特征)带有较少的信息，而出现频率较少的词条带有较多的信息。TF-IDF 的值表示权重，词条界T_j文档d_i中的 TF-IDF 值由下式定义：

$$w_{ij}=TF_i\times\log(N/DF_i) \quad (2.3)$$

其中，TF_i是词条T_j在文档d_i中出现的次数；DF_i表示整个文档集D中包含

词条T_j的文档数，称为文档频率，IDF_i为DF_i的倒数，称为倒排文档频率;N 表示统计语料中的文档总数。因此，文档d_i可以表示成一个特征向量:

$$d_i = \{w_{i1},...w_{ij},...w_{in}\} \quad (2.4)$$

两个文档d_i和d_j的相关度等于两个文档对应的特征向量的内积，公式如下:

$$\mathrm{Sim}(d_i,d_j) = \frac{\sum_{k=1}^{M} W_{ik} \times W_{jk}}{\sqrt{(\sum_{k=1}^{M} W_{ik}^2)(\sum_{k=1}^{M} W_{jk}^2)}} \quad (2.5)$$

其中，d_i，d_j为文本的特征向量，M 为特征向量的维数，W_k 为向量的第 K 维。

2.3.3 爬行策略

主题搜索引擎提供主题领域内的信息查询，非主题领域内的信息对其而言是无效信息。这就要求主题搜索网络爬虫在进行网上信息采集时，必须采用主题搜索策略按照预先规定的主题去搜索网上相关信息，从而达到既提高索引数据库中的信息质量又有效减少搜索工作量的目的。目前比较有代表性的主题搜索策略简介如下:

✧ 网页重要度优先搜索(Best First)

J.Cho 等人[16]提出基于网页重要性优先的主题搜索策略，列举出几种网页重要性评价指标:网页内容相似性、网页的入度、网页的出度、网页的 PageRank 值和 URL 提示信息等。本文所设计的系统的搜索策略采用该方法。

✧ Fish 搜索算法

P.DeBra 等人[17]首次提出“Fish”搜索算法。主题搜索网络爬虫动态维护一个按搜索优先权值排序的未搜集 URL 列表，并根据它选择下一步搜集目标。在信息搜索过程中，相关网页包含的超文本链接被赋予比不相关网页包含的超文本链接更高的优先权值，插入到未搜索 URL 列表中。

✧ SharkSearch 搜索算法

M.Hersovic 等人[18]在“FishSearch”算法基础进行改进提出了“SharkSearch”算法。该算法在计算 URL 的优先权时考虑了超文本链接描述文

字的提示作用，并且采用向量空间模型 VSM(Vector Space Model)计算网页的相似度，细化了搜索优先权值的计算。

✧ 基于神经网络搜索

F.Menczer 等人[19]设计的 InfoSpider 引入神经网络，通过抽取网页超文本链接提示信息作为神经网络的输入，将其输出结果作为进一步选择搜索超文本链接的依据。被搜索网页的相关度作为反馈用于训练神经网络。

第三章 系统关键技术

现今对深度网的研究越来越多。据统计，目前深度网所包含的信息量是普通搜索引擎可收录信息量的 500 倍并且这些信息大多为结构化的高质量数据。不仅如此，深度网资源所涉及的领域非常广泛，因此对于用户所需要查找的一些专题领域的要求可以提供较好的信息资源。基于此，对深度网的研究具有重大的现实意义和理论价值。

本文所设计的爬虫在爬行主题页面的同时能够较好的爬行深度网资源。本章首先介绍深度网的概念；其次阐述目前对深度网研究的一些相关技术；最后给出本文所设计的系统的几个关键技术问题的解决方法。

第一节 深度网的介绍

浅层网（Surface Web），或可见网，是 Web 中可以通过通用 Web 搜索引擎访问到的部分。可以通过几乎所有的主题目录访问浅层网。深度网（Deep Web）或不可见网，是 Web 中不能通过搜索引擎这类工具检索到的部分[20]。深度网资源通常为 Web 在线数据库

随着 World Wide Web 的飞速发展，出现了越来越多的可以在线访问的数据库。据统计，目前 Web 数据库的数量已经超过了 45 万个。在此基础上构成了大量深度网资源。Deep Web 蕴含了大量有用的信息，其价值远远超过了仅由网页构成的 Surface Web。由于深度网资源蕴藏海量信息并且这些信息多为结构化或半结构化的高质量数据，因此对深度网的研究具有重大意义。

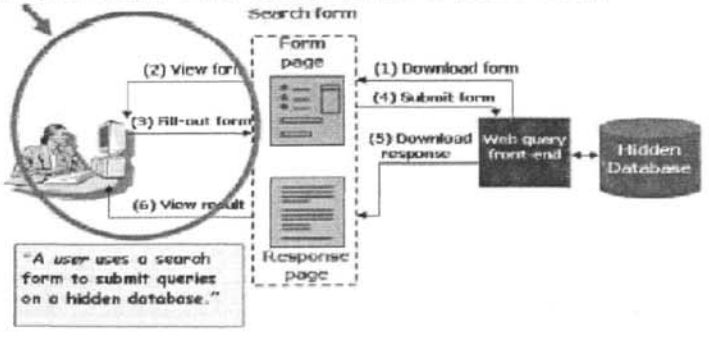

图 3.1 深度网资源的用户访问过程[21]

但由于对 Web 数据库的访问只能通过其提供的查询接口，因此很难被一般的搜索引擎获取到。由于 Deep Web 的大规模性、动态性以及异质性等特点，通过手工方式远远不能在效果和效率上满足用户对信息获取的需要。为了帮助人们快速、准确地利用 Deep Web 中的海量信息，在对 Deep Web 数据集成方面的研究逐渐成为这一领域的一个研究热点。研究者力图提出一种通用的方法，实现对现实世界各个领域的 Deep Web 数据的集成,并在查询接口集成和数据抽取等方面已经取得了实质性的进展。图 3.1 为用户访问深度网资源的过程。图 3.2 为深度网爬虫访问过程。对比两图，可以看出深度网爬虫在访问过程中模拟了用户访问的方法，对表单模拟提交查询并分析返回的结果。

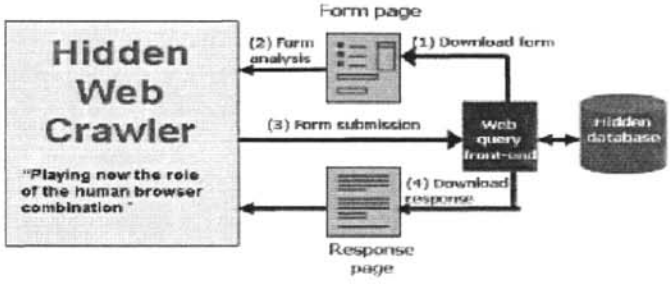

图 3.2 深度网爬虫的爬行过程[21]

最早的深度网资源出现在 2000 年，这给当时 Web 搜索引擎工作者提出一项崭新的具有挑战性的课题。从那时起，对深度网的研究成为搜索引擎工作者的一个重要方向。到目前为止，一些爬虫和索引工具已经能够解决一些爬行深度网遇到的技术难题。

目前大多数搜索引擎可以发现的深度网资源包括：

1. 非 html 格式的页面（包括 PDF, Word, Excel, PowerPoint 等）。

2. 基于脚本的页面，即 URL 中包含？或其它脚本代码。

3. 由某些数据库软件动态生成的页面（例如，ASP, Cold Fusion）。如果存在爬虫可以爬行的固定的 URL 地址，则这些页面可以被索引。

第二节 深度网爬行研究的关键问题及相关技术

深度网资源覆盖了很多领域，从专业信息的查询（如图书信息检索）到日常生活资料查询（如旅店信息查询）。不仅如此，深度网所蕴含的数据资源多为结构化或半结构化数据，质量高，这些都使得对深度网的研究越来越备受关注。

由于深度网资源所涉及的范围广、信息质量高、资源巨大，所以需要有一种高效的便捷的方法来利用这类资源。对深度网资源的集成便是深度网领域的一项重要应用。

然而对深度网资源的集成面临着很大的挑战[3]：

1．深度网资源的规模大约为静态页面资源[22]的 400 到 500 倍，并且仍在快速增长。

2．查询条件差别较大，几乎没有规则来支持建立统一的查询入口。

基于这样的问题，对深度网的研究主要集中在对深度网表单查询入口的定位和查询入口模式的抽取两方面。

3.2.1 深度网表单查询入口的定位

针对深度网表单查询入口的定位，目前的方法大多基于对表单页面的定位。L. Barbosa 在[1]中提出一种聚焦表单的深度网爬虫模型，FFC(Form-Focused Crawler)。FFC 的体系架构如图 3.3 所示。

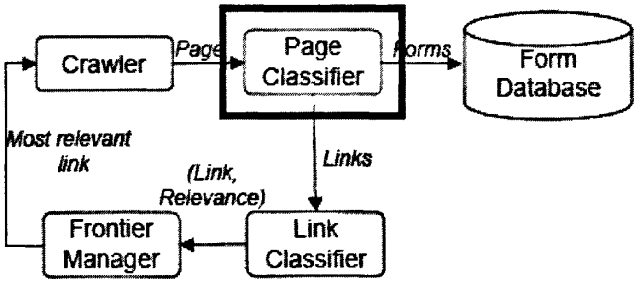

图 3.3 FFC 体系结构[1]

FFC 建立的思想是：采用链接分类器 Link Classifier 来识别和维护一组 URL 链接地址，这组 URL 地址通常可以链接到表单入口页面。分类器按照距离表单入口页面的链接步数的大小来排序这些 URL 链接，距离近的优先级高。分类

器由一组包含深度网表单入口的页面来训练得到。FFC 的实验结果显示，对大量网页进行爬行可以较有效地发现深度网表单入口。但是，FFC 有其局限性，主要体现在：

1. 对于链接分类器的建立和识别特征的选取需要大量的人工参与。

2. 实验结果高度依赖于分类器的训练样本。

3. 结果获得的深度网表单入口的结构各不相同，甚至包含不同领域的表单入口页面。

经过对 FFC 的分析和总结，L. Barbosa 等人又在此基础上，在[2]中提出一种适应性的深度网表单入口的定位方法, ACHE(Adaptive Crawler for Hidden-Web Entries)。ACHE 爬虫总体框架如图 3.4。

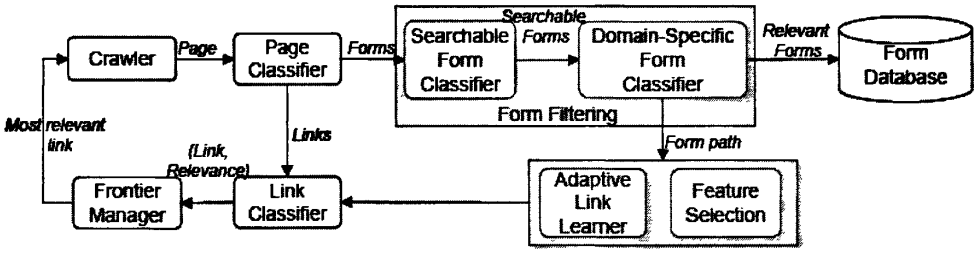

图 3.4 ACHE 架构[2]

图 3.4 中白色的模块就是 FFC 爬虫的架构，在此基础上 ACHE 增加了蓝色的模块，包括 Domain-Specific Form Classifier、Adaptive Link Learner 和 Feature Selection 模块。

Adaptive Link Learner:在 PPC 中链接分类器 Link Classifier 是通过线下学习的方式得到。通过对一组选定的包含表单入口的页面样本进行学习，从中选取这些页面的 URL 链接格式的规则。以此作为判别 URL 链接是否可以在有限步内链入表单入口页面的标准。而 AHCE 中，Adaptive Link Learner 采用爬行过程中获得的表单入口页面的 URL 地址特征为标准。通过定期对爬行过程中成功发现的深度网表单入口的 URL 地址提取规则，生成新的分类器，以更新原有的 Link Classifier。

Feature Selection:特征选择是伴随适应性链接学习器 Adaptive Link Learner 同时应用的。用于抽取出指向深度网表单入口的 URL 链接及其上下文的特征。

Domain-Specific Form Classifier:特定领域表单分类器使用HIFI（Hierarchical classification framework for Identifying Forms Interface in a domain）来过滤不相关的表单。HIFI采用两层分类器过滤表单，GFC(Generic Form Classifier)和DSFC(Domain-Specific Form Classifier)，如图3.5所示。

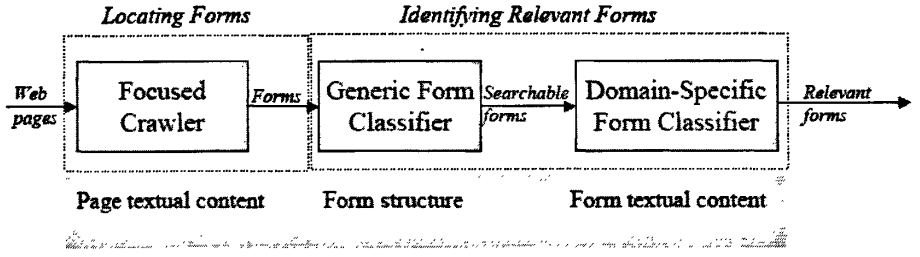

图 3.5 HIFI

Generic Form Classifier 用于过滤出不可查询的表单，如登录表单、电子邮件等。Domain-Specific Form Classifier 过滤出与主题不相关的表单，只保留相关表单。

此外，也有一些深度网集成的系统采用其它的方式进行表单入口的定位,如MetaQuerier[3]。MetaQuerier 是一个大型的深度网数据集成系统，其中包含深度网在线数据库定位。与其它的定位方法不同,MetaQuerier 既不聚焦主题页面，也不根据 URL 链接预测表单入口页面。它基于对大量查询表单位置的观察，发现这些表单通常位于网站根目录或距根目录很近的目录上。因此，MetaQuerier 以一组有效的 Web 服务器的 IP 地址作为种子，然后从这些服务器的根目录页面起，以广度优先爬行固定深度的页面，以获取查询表单。

通过分析，FFC 和 ACHE 的深度网入口定位方法主要用于在深度网的集成应用当中。这种应用要求爬虫只关心存在入口表单的页面，因此爬虫通常要对一定量含查询表单的页面学习它们的 URL 地址中的规则，以便在实际爬行中预测是否将要爬行的 URL 地址会包含查询表单。对于本文所设计的系统来说，在深度网入口定位的问题上并不是对未爬行的页面预测其是否包含查询表单，更多的是对出现表单入口的主题相关页面上判别其表单是否属于与领域相关。这依赖于预先选定的本体域，通过与本体域的相关程度判别表单是否为所需。这部分将在本章第三节详细介绍。

3.2.2 深度网表单查询入口模式（schema）的抽取

传统的模式匹配通常是在人工条件下进行。但是随着 Web 的飞速发展，人工模式无法完成如此庞大的工作量。目前，自动化或半自动化的模式匹配系统已经得到广泛的研究。比较有代表性的系统，包括：微软的 Cupid 系统[23]、Stanford 大学的相似度洪泛方法(Similarity Flooding method)[24]、Washington 大学的 GLUE 和 LSD 方法[25][26]等等。

Cupid 系统根据 schema 中元素的名字、数据类型、约束以及 schema 的结构来匹配 schema 中的元素。他们通过计算不同 schema 中元素之间的相似系数，然后通过这个系数来推出两个 schema 之间元素的匹配关系。

这个系数的计算分为两个部分：语言相似系数和结构匹配系数。

语言相似系数是通过 schema 中元素的名字，数据类型，以及领域信息等等来匹配两个 schema 中每个元素。在识别缩写简写以及同义词时，通常使用词典来辅助完成，从而匹配两个 schema 元素的名字，得到在两个 schema 中每对元素之间的语言相似系数。

结构匹配系数是根据这些 schema 元素的内容和邻近关系来计算每对 schema 之间的结构匹配程度。

最后，通过给出权重把这两个所得系数结合在一起，得到了一个相似系数，并把具有相似系数最大的每对元素匹配在一起。这种方法通常用于 1：1 关系的匹配，适合于大多数查询条件简单的表单入口的匹配。

Sergery Melnik 等人在[24]中提出了一种半自动化的匹配算法，Similarity Flooding Algorithm。该方法用于匹配不同的数据结构（或称为模型 model），包扩数据模式(data schemas)、数据实例(data instances)或两者的混合。模型的元素通常代表某一属性，如关系表的字段等。算法遵循的思想：

1.将待匹配的一组模型 model 转换为一组有向标签图(Directed Labeled Graphs)。对转换后的 DLG 图采用一种迭代的 fixpoint 计算的方法来获取相似结点，即获得两个图中的相似的结点。对于相似度的计算，Sergery Melnik 等人采用了一种迭代的方法。该方法基于这样的假设：对于不同模型中的两个元素，如果它们的邻接元素是相似的，则它们也认为相似。换句话说，假设 A 和 B 两个模型，a 是 A 中的元素，b、c、d 是 A 中与 a 邻接的元素；e 是 B 中的元素，f、g、h 是 B 中与 e 邻接的元素，若 b、c、d 与 f、g、h 是相似的，则 a 与 e 相似。

这种相似度传播的匹配方法使人很容易联想到网络中 IP 包广播通信的方法，因此这一算法称为相似度洪泛方法。

2. 对于第一步得到的结果，称为映射（mapping）。根据特定的匹配目标对结果映射进行过滤，得到它的一个子集作为最终结果。

以匹配关系模式 S1 和 S2 为例解释该算法。图 3.6 为两个表示人员或员工的关系表结构。

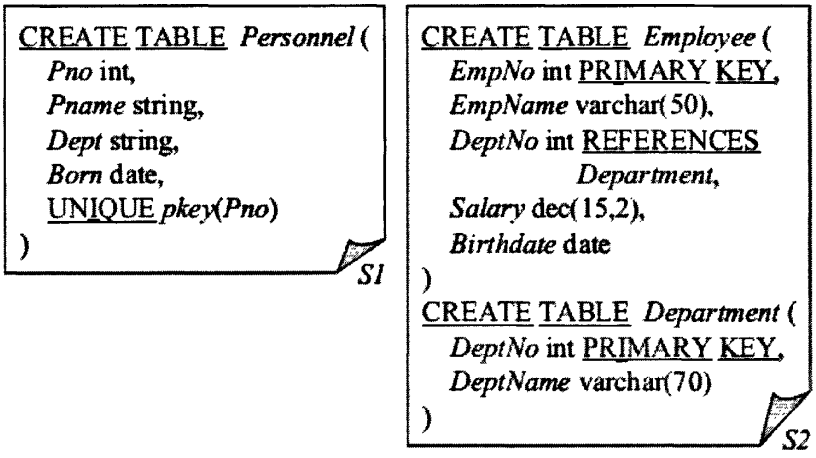

图 3.6 匹配两个关系模式：Personnel 和 Employee-Department

按照 Similarity Flooding 算法，将关系模式 S1、S2 转换成 DLG 图。图 3.7 出示为 S1 转换得到的 DLG 图 G1。

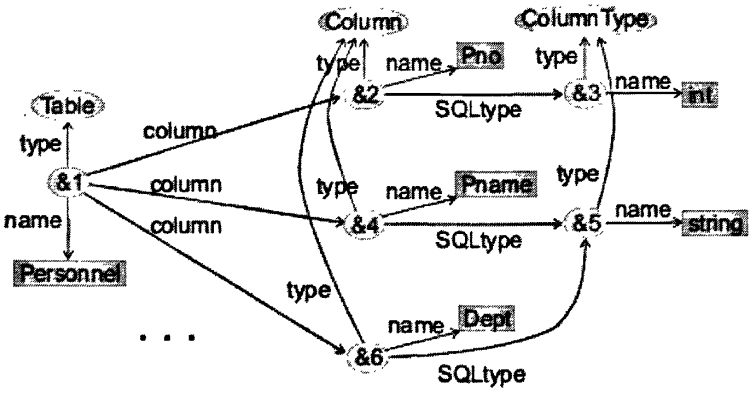

图 3.7 由关系模式 S1 获得的 DLG 图 G1

对 G1, G2 采用 fixpoint 计算方法得到一组初始的 mapping，如图 3.8 所示。

Line#	Similarity	Node in G_1	Node in G_2
1.	1.0	Column	Column
2.	0.66	ColumnType	Column
3.	0.66	'Dept'	'DeptNo'
4.	0.66	'Dept'	'DeptName'
5.	0.5	UniqueKey	PrimaryKey
6.	0.26	'Pname'	'DeptName'
7.	0.26	'Pname'	'EmpName'
8.	0.22	'date'	'Birthdate'
9.	0.11	'Dept'	'Department'
10.	0.06	'int'	'Department'

图 3.8 初始 Mapping(前 10 行)

根据 Similarity Flooding 算法的第二步，对初始 mapping 过滤得到最终结果，如图 3.9 所示。

Similarity	Node in G_1	Node in G_2
1.0	Column	Column
0.81	[Table: Personnel]	[Table: Employee]
0.66	ColumnType	ColumnType
0.44	[ColumnType: int]	[ColumnType: int]
0.43	Table	Table
0.35	[ColumnType: date]	[ColumnType: date]
0.29	[UniqueKey: perskey]	[PrimaryKey: on EmpNo]
0.28	[Col: Personnel/Dept]	[Col: Department/DeptName]
0.25	[Col: Personnel/Pno]	[Col: Employee/EmpNo]
0.19	UniqueKey	PrimaryKey
0.18	[Col: Personnel/Pname]	[Col: Employee/EmpName]
0.17	[Col: Personnel/Born]	[Col: Employee/Birthdate]

图 3.9 过滤后的结果

第三节 系统关键技术与方法

针对本章前两节提出的深度网入口的定位和模式抽取以及主题爬虫的特征词训练的问题，本节给出以下三个方法解决上述问题。

3.3.1 在线主题特征词学习算法

垂直搜索引擎与通用搜索引擎最大的区别在于垂直搜索引擎是面向某个领域的。这就要求垂直搜索引擎的爬虫在爬行过程中只关心与主题相关的网页，对于主题无关的网页要予以排除。完成这项工作的爬虫就是主题爬虫。

相关度判别就是计算网页与主题领域的相关程度。通过将已爬行的网页表示成一个页面特征词向量，再与主题向量计算内积从而得到该页面相对于主题向量的相关度。流程图如图 3. 10 所示。

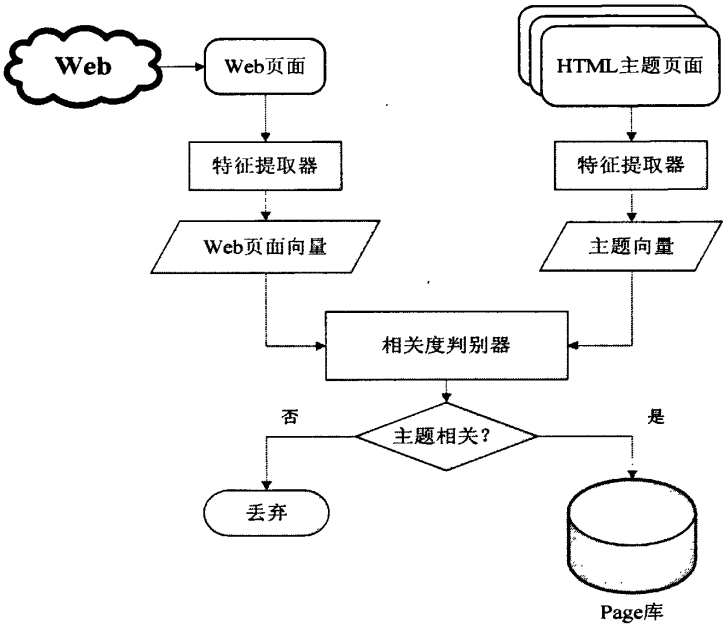

图 3.10 相关度判别

由图 3.10 所示，HTML 主题页面是预先选定的一组表现主题的页面，通常由该领域的专家进行选取或网页分类目录选取的种子网站，如 Yahoo!分类目录。利用特征提取器分别对主题页面和已爬行的 Web 页面提取特征生成向量，然后通过相关度判别器进行判断。

对于给定的一组样本，本文采用一种基于文档频率的在线关键词学习的方法来提取主题关键词并适当调整。这一方法需要用到对既定样本特征的提取和对抓取的页面相关度的计算。下面分别介绍。

3.3.1.1 特征提取

在相关度计算中，较为常见的通常有两种模型：布尔模型和向量空间模型。基于布尔模型的相关度计算实现简单，通常是预先选定一组特征词来代表一类文档。对于某一篇文档，如果出现某一特征词，则该文档对于这个出现的特征词用 1 来表示，否则用 0 来表示。在计算两篇文档的相关度时，就取这两篇文档所表示的特征词的交集，交集特征词越多，说明这两篇文档相关度越高。

目前使用最多的是基于向量空间模型（VSM）的相关度计算方法。salton 等人在[14]中提出了向量空间模型算法。向量空间模型的基本思想是以向量来表示文本：$D=\{w_1,w_2,...,w_n\}$，其中 w_i 为第 i 个特征项的权重。对于特征项的选取，一般可以选择字、词或词组，根据实验结果，普遍认为选取词作为特征项要优于字和词组，因此，要将文本表示为向量空间中的一个向量，就首先要将文本分词，由这些词作为向量的维数来表示文本。

对文档的分词是特征提取的首要问题。现有的分词算法可分为三大类：基于字符串匹配的分词方法、基于理解的分词方法和基于统计的分词方法。当前在分词领域的研究中大多不是以某一种方法来分词，而是将不同的方法结合在一起。在本文设计的系统中，对中文分词采用了现有的分词组件。

目前在中文分词研究领域已出现许多较为成熟的产品。如中科院计算所的汉语词法分析系统 ICTCLAS、海量智能分词系统、CSW 中文智能分词组件等。ICTCLAS(Institute of Computing Technology, Chinese Lexical Analysis System) 是中国科学院计算技术研究所在多年研究基础上，耗时一年研制出的基于多层隐马模型的汉语词法分析系统，该系统的功能有：中文分词；词性标注；未登录词识别。分词正确率高达 97.58%(973 专家组评测结果)，基于角色标注的未登录词识别能取得高于 90%召回率，其中中国人名的识别召回率接近 98%，分词和词性标注处理速度为 31.5KB/s。CSW 中文智能分词 DLL 组件，可将一段文本自动的按常规汉语词组进行拆分,并以指定方式进行分隔，且可对其拆分后的词组进行语义、词频标注，广范应用于各行各业的信息资料检索、分析。目前业界评论的国内最好的中文分词技术是海量科技的智能分词系统，其分词准确度超过 99%，由此也使得中搜在搜索结果中的错误率很低。海量智能计算技术研究中心为了使中文信息处理领域的研究者们能够共同分享海量智能中心的研究成果还发布了海量智能分词研究版，供专家、学者和爱好者进行研究。

在本文所设计的系统中，采用了目前广泛使用的 ICTCLAS 3.0 组件作为特

征提取器的分词工具。该组件在分词精度与分析速度上都有重大的突破，API 不超过 200KB，各种词典数据压缩后不到 3M，目前已经支持 C/C++/C#/Delphi/Java 等主流的开发语言。

对于文档向量 $D=\{w_1,w_2,...,w_n\}$，由于它的每一维 w_i 是第 i 个特征词的权重，所以在分词获得每一维的特征词后还要对其计算权重，即该词对主题的共献或评价。经常使用的特征提取的评价函数有文本频率(document frequency, DF)、chi-square (CHI)、信息增益(information gain, IG)、互信息(mutual information, MI)、term strength(TS)、GSS Coefficient、odds ratio 等[27]。Yang 等在 Reuters21578 语料库上试验了前面 5 种方法，认为 DF、CHI、IG 更为有效[28][29]。国内的有些学者则认为 MI>DF>IG[30]。

对于词权重 W 的计算来说，使用最多的是TF/IDF方法，TF(Term Frequency)、IDF(Inverse Document Frequency)为关键词的词频与倒排文档频率。对于文档j中的关键词i来说，则 TF_{ij} 表示特征词i在文档j中的词频。DF_i (Document Frequency)表示所有文档集合中出现特征i的文档数目。IDF_i 的计算公式为 $\log(N/DF_i)$，其中 N 为所有文档的总数。权重 W 的计算公式为：

$$W_i = TF_i * IDF_i = TF_i * \log(N/DF_i) \quad (3.1)$$

本文采用一种基于文档频率的在线关键词学习的方法来提取主题关键词并适当调整，以求达到较好地区别主题的能力。算法分线下学习和在线学习两部分。通过线下学习来获取一组带权重的主题特征词并通过在线学习进行特征词的权值调整以及特征词的更新。线下学习步骤如下：

1. 预先选取可以表现某一主题的一组网页集合 $P=\{page_1,page_2,...\}$，$N=|P|$ 表示网页集P中的网页数量。

2. 从 P 中取出 $page_1$，利用ICTCLAS组件对其分词获得一组词集 $I=\{i_1,i_2,...,i_n\}$。对词集 I 中的每一个词 i 统计其在网页中出现的次数，即词频 TF_i，并根据公式 $IDF_i=\log(N/DF_i)$ 计算该词的 IDF_i。此时，由于只处理了 P 中的一个网页，所以对于每个词，DF_i 都等于1，即 $IDF_i=\log(N)$。对每个词按照公式(3.1)计算权重，然后将词按权重排序，选择权重大于某一阈值的词加入到主题特征词集作为主题词使用。如果该词在主题词集中不存在则直接加入，否则不加。

3. 重复步骤2，直到 P 集合中的所有页面都处理完毕。

4. 每新加主题特征词后都要对原主题特征词集中的关键词的权重进行调整。调整的规则为，对于在文档 j 中的词 i，如果词 i 已经存在于主题词表中，则该词 i 的词频等于原来的词频与其在文档 j 中的词频之和，原词表中的 DF_i 增1；若词 i 不在主题词表中，则 i 的词频等于它在文档 j 中的词频，DF_i 等于1。对调整后的词频和 DF_i 重新按照公式(3.1)计算权重。

5. 对最终的特征词集进行归一化处理。

通过上述算法，可以获得一组带权重的主题特征词集。在进行爬行的过程中，可以开启在线学习功能，该功能为可选，过程如下：

1. 对于URL库中待爬行的链接进行爬行，并进行相关度计算。对于相关度大于某一阈值 σ 的页面认为是主题页面。

2. 对判定为主题页面的网页进行关键词的学习，学习的过程同线下学习。

3. 调整学习后的主题特征词集及权值。

通过在线学习算法，可以对主题词集进行调整及关键词更新。

这里值得注意的一点是，在爬行过程中认为相关度大于阈值 ω 的页面仅为主题相关，而只有相关度大于阈值 σ 的页面才为主题页面，可以进行关键词学习使用。这里 σ 值要大于 ω 值。这样处理基于的假设是：只有非常相关的页面才可以作为关键词学习使用，以防止主题特征值的漂移。

3.3.1.2 相关度计算

提取主题相关的网页需要有一种主题相关度计算的方法。通常这种计算主题相关度的方法有一个阈值。当对网页计算出的结果大于阈值则被认为是主题相关，否则认为主题无关。主题相关度计算分两部分，一是对已下载的网页进行相关度判断；二是对网页中的超链接进行主题相关度预测。

主题爬虫将网页下载到本地后，需要使用基于内容的主题判别方法计算该网页的主题相关度值，主题相关度低于某一阀值 ω 的网页被丢弃。如果该网页的主题相关度高于阈值，则认为是相关的，进行保存，然后解析网页中向外的超链接并根据链接的上下文预测超链接所指的网页的主题相关度。对于链接的预测有两种结果，一是预测认为相关，则将链接地址存入待爬行的 URL 队列等待爬行；二是预测认为无关的，将其存入待爬队列并对其标记一个参数。这个参数代表爬行无关网页的当前层数，即当前的爬行深度。当它大于预先设定的爬行深度值则丢弃该链接。这种技术称为隧道技术(Tunneling) 。Bergmark 在[31]

提出了隧道技术，这是一种启发式的全局最优算法，使用隧道技术的爬虫爬行到无关网页时并不停止，而是继续往这个路径上向前探索 K 步，K 的大小由人工设定。使用隧道技术的前提是基于网络上存在 Web Community[32]。Web Community 是一组主题相似的网页，WWW 上的网页通常以 Web Community 的形式存在。但主题相似的网页可能分别在不同的 Web Community 中，中间通过无关网页连接。这样爬虫在爬行到无关网页时可以继续向前探索，只要不同 Web Community 间的路径不非常远，就可以到达另一个 Web Community 以获取更多的主题相关网页信息。

网页经过分词程序处理后，首先去除停用词，合并数字和人名等词汇，然后统计词频，计算权重。对于权重的计算，除(3.1)的公式以外，还存在多种 TF/IDF 公式，本文设计的系统中对爬行过的网页采用了一种比较普遍的 TF/IDF 公式：

$$W_{ij}=\frac{TF_{ij}*\log(N/DF_i)}{\sqrt{\sum_k[TF_{kj}*\log(N/DF_k)]^2}} \quad (3.2)$$

其中，分母为归一化因子，分子各部分意义同(3.1)。

公式(3.2)实际上是对公式(3.1)的归一化处理。在上节所述的学习主题特征词的方法中，最后也要对主题词词权做归一化处理。

最终将网页内容的文本表示为上面描述的向量。文本描述为向量后，就要进行相似度的计算，公式如下：

$$\mathrm{Sim}(d_i,d_j)=\frac{\sum_{k=1}^{M}W_{ik}\times W_{jk}}{\sqrt{(\sum_{k=1}^{M}W_{ik}^2)(\sum_{k=1}^{M}W_{jk}^2)}} \quad (3.3)$$

其中，d_i，d_j为文本的特征向量，M 为特征向量的维数，W_k 为向量的第 K 维。

在实际操作中，d_i和d_j一个是待判别的文本向量，一个是主题特征向量。

在相关度判别中，除对已下载的页面进行相关度的计算，还要对从页面中解析出的链接进行相关度的预测。目前对网页锚链接相关度预测的方法通常有基于父网页的主题相关度预测、基于链入网页的主题相关度预测和 TPR 主题相关度预测。本文设计的系统采用基于父网页的主题相关度预测方法。

对于锚链接相关度的预测主要依赖于锚链接的文本。锚文本通常代表对锚

链接所指向的网页的评价和描述。因此，锚文本可以为它指向的目标网页的主题预测提供非常重要的信息[33]。由于锚文本在某些时候不能很好地表达意义，例如，锚文本可以是“详细信息”或“更多”等。这时候通常取锚文本的上下文代替链接的意义。对锚文本按照（3.4）计算相关度。

$$Sim(AnchorText)=\sum_{i=1}^{n}V_t * T，（3.4）$$

其中，V_t为锚文本关键字向量，T为主题向量。

对于两个锚文本的相关度预测值相同的情况，如果包含该锚链接的页面主题相关度较高，则该锚链接指向的页面主题相关的概率也较大。基于这一现象，通常父网页的相关度也对锚链接的相关度预测起到一定作用。因此，基于父网页的主题相关度预测计算公式调整为：

$$Sim(Anchor)=(1-\lambda)*\sqrt{\sum_{i=1}^{N}Sim(P_i)^2}+\lambda * Sim(AnchorText)\ (3.5)$$

其中，$Sim(Anchor)$为目标锚链接的相关度预测值；$Sim(P_i)$为目标锚链接的第 i 个父网页的主题相关度；$Sim(AnchorText)$为对锚链接包含的文本的主题相关度；λ为影响系数，通常在 0 到 1 之间。

由公式(3.5)可以看出，λ越小，对于目标网页的相关度预测越取决于父节点网页；λ越大，相关度预测越取决于锚文本自身的相关度。

3.3.2 基于本体域的深度网入口定位方法

本文提到的基于本体域的深度网入口定位方法本质上是一种主题表单判别方法。总体思想是对获取的可查询形式的表单判别该表单是否属于想要查询的领域。直观的例子如图 3.11 和图 3.12 所示。

图 3.11 药品查询表单(drug.39.net)

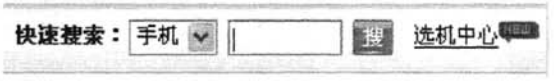

图 3.12 手机查询表单(mobile.sina.com.cn)

由图 3.11 和图 3.12 所示，如果所选则查询的领域是医药类，那爬虫必须丢弃图 3.12 的表单而选择图 3.11 的表单。所以爬虫要做的工作就是判别哪个表单是属于所要查询的领域的。对于这种主题表单的定位方法需要依赖于代表某一主题的本体域以及表单相对于本体域而言的相关度。以下分别介绍。

3.3.2.1 本体域的定义

本体是一个源于哲学的概念，原意是指关于存在及其本质和规律的学说，后来引入人工智能领域，特指对概念化的一个显式的规格说明(explicit specification of conceptualization)。对本体的定义有很多，最早给出本体定义的是Neches[34]等人，他们将本体定义为“给出构成相关领域词汇的基本术语和关系，以及利用这些术语和关系构成的规定这些词汇外延的规则的定义”。其中得到广泛认可的是Studer[35]在Gruber 和Borst 的定义基础上提出的“本体是共享概念模型的明确的形式化规范说明”。这包含四层含义:a. 概念模型(conceptualization)，指通过抽象出客观世界中一些现象(phenomenon) 的相关概念而得到的模型。概念模型所表现的含义独立于具体的环境状态。b. 明确(explicit)，指所使用的概念及使用这些概念的约束都有明确的定义。c. 形式化(formal)，指本体是计算机可读的(即能被计算机处理)。d. 共享(share)，指本

体中体现的是共同认可的知识，反映的是相关领域中公认的概念集，即本体针对的是团体而非个体的共识。从定义上来看，本体作为一种能在语义和知识层次上描述概念体系的有效工具，与图书馆中的规范化的词表如主题词表、叙词表有着很大的相似之处。

本体域的定义对判别入口表单是否与主题相关有着重要作用，可以帮助我们较好地识别我们所要找的表单。在本系统中，我们将遵循 Studer 的四层定义将本体域定义为一组属性的集合。

定义 $A=\{a_1,a_2,...,a_n\}$ 为一组属性。对于 A 中的每一个属性 a_i，都有一个属性名 a_i_title 和一组别名 $Alias=\{a_i\_alias_1,a_i\_alias_2,...,a_i\_alias_n\}$ 以及一个属性参数 s_i。该参数用于描述该属性在本体域中的重要程度，一般取值在 $(0,1]$。对于 s_i 值没有标准的计算方法，通常根据经验给定。例如，批准文号只有在药品类商品中才会存在，因此对于药品查询的本体域，批准文号的 s_i 值就可选择为 1。在本系统中，所要处理的查询入口都是针对药品的查询，因此，对本体域属性集的定义基本上都是药品的相关信息参数，定义如下。

表 3.1 药品查询表单的本体域定义

属性	属性名	别名	参数
a_1	药品名称	名称，品名，药名	0.7
a_2	药品编码	编码	0.9
a_3	英文名称	英文名	0.05
a_4	生产企业	生产厂家，厂家	0.05
a_5	批准文号	批号	1
a_6	药品分类	分类，类别	0.6
a_7	助记码		0.05
a_8	处方		1
a_9	药物成分	成分	0.6
a_{10}	药效	疗效，功效，效果	1
a_{11}	适应症	适用症	0.8
a_{12}	功能与主治	功能，主治	0.9
a_{13}	GMP 达标		1
a_{14}	OTC		1

由表 3.1 可以看出，本体域中定义的属性基本上是药品的相关信息。与药品相关的信息有很多，远远超过上表中的定义，还包括化学名称、通用名称、剂型、有效期、禁忌症、形状、药理毒理和药代动力学等等一系列的信息。但在实际操作中，查询表单不会给用户提供过于专业的条件，所以在本体域定义中只定义一些用户常用的信息作为属性。这一点与普遍意义上的本体域定义是有区别的，后者定义的信息量要大得多。

属性用属性名来表示，别名是属性名的一种替代标签，用于在表单中发现属性，即如果查询表单中出现药名标签，那么它实际上说的是a_1药品名称属性。除了给本体域定义一组属性之外，还需要设定一个阈值χ，用于表示查询表单与本体域的相关程度。高于阈值，则认为是本体相关，否则，认为是本体无关的，不进行处理。

3.3.2.2　查询表单领域相关度判别

这里假定一个表单可以抽取出一组属性组成表单的属性集合并由该属性集合来表示表单。定义一个表单的属性集合$F_A=\{f_1,f_2,...,f_n\}$。对于集合中每一个属性f_i，定义一个四元组。$f_i=\{f_e,f_t,a,\upsilon,W\}$。$f_e$表示表单中的标签元素的内容；$f_t$表示对应该标签元素的文本信息；$a$表示该标签元素在本体域中对应的属性；$\upsilon$为该属性对表单而言的信任度或说贡献，在本文中为简单起见，将信任度设为一个常量 1；W表示该属性可能的查询值。

对于表单属性的抽取将在 3.3.3 节详细介绍。本节主要介绍抽取属性后的表单如何与本体域进行匹配。实际上，对表单抽取进而得到的属性集合与本体域属性集合间的相关度计算可以回归到两个特征向量间的相关度计算的方法，即计算向量$F_A=\{f_1,f_2,...,f_n\}$与向量$A=\{a_1,a_2,...,a_n\}$之间的相关度。所不同的是，对于特征向量中各个维的值来说，表单特征向量F_A取各属性的信任度υ，而本体域向量A取本体域各属性的s参数。公式如下：

$$Sim(Form)=\sum_{i=1,j=1}^{n}s_i*\upsilon_j \quad (3.6)$$，式中，s_i是本体域中属性 i 的参数，υ_j是F_A中相对应的属性的信任度。

当$Sim(Form)$的值大于某一给定的阈值χ时，认为该表单是该领域的表单，可以进行后续的查询工作。

下面举例说明表单的判别过程。假设阈值χ=2。

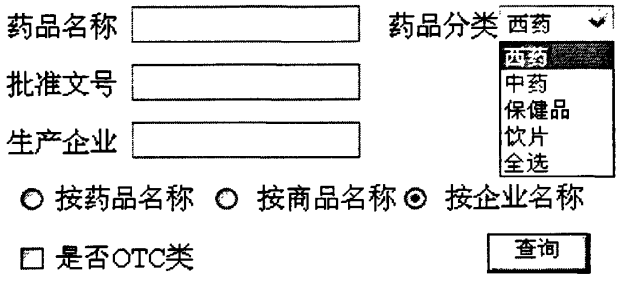

图 3.13 表单查询入口

例如，对图 3.13 所示的表单查询入口进行属性抽取后得到的属性集合如下所示。$F_A=\{f_1,f_2,f_3,f_4,f_5,f_6,f_7,f_8\}$

f_1={<input type="text" size="16" name="med_name">, “药品名称”, a_1, 1, “阿斯匹林、复方甘草片”}

f_2={<input type="text" size="16" name=" med_no ">, “批准文号”, a_5, 1, “”}

f_3 ={<select size="1" name="f1"><option value="西药">西药</option><option value="中药">中药</option><option value="保健品">保健品</option> <option value="饮片">饮片</option><option value="" selected>全选</option></select>, “药品分类”, a_6, 1, “保健品、西药、中药、饮片”}

f_4={<input type="text" size="16" name="mcnr">, “生产企业”, a_4, 1, “隆顺榕、天士力”}

f_5={<input type="radio" value="ym" name="mclb">,“按药品名称”, , 1, “”}

f_6={<input type="radio" value="sm" name="mclb">,“按商品名称”, , 1, “”}

f_7={<input type="radio" value="qm" name="mclb">,“按企业名称”, , 1, “”}

f_8 ={<input type="checkbox" value="OTC" name="otc">, “是否 OTC 类”, a_{14}, 1, “是、否”}

根据表 3.1 设定的本体域按照公式(3.6)计算：

$Sim(F_A)=a_1\times\upsilon_1+a_5\times\upsilon_2+a_4\times\upsilon_3+a_{14}\times\upsilon_8$ =0.7+1+0.05+1=2.75>2，所以该表单属于本领域的表单。

3.3.3 基于网页标签元素距离和同义词比较的深度网入口模式抽取方法

本节将讨论如何从表单中抽取属性。属性的抽取基于网页中标签元素间的距离和与本体域中词义的比较来实现。在介绍具体抽取方法之前要引入同义词语义比较的概念。

3.3.3.1 同义词语义比较

同义词语义的比较就是判断两个词语的语义是否是同义或从广义的角度上是否是同类。对于表单属性的抽取而言，我们要进行表单属性与本体域属性的匹配。然而，表单上属性的名称或表示方法常常因网页设计者的风格不同而多种多样，本体域无法穷尽这些名称，所以将表单属性的名称与本体属性的名称及别名的一种语义上的比较往往非常必要。

同义词比较涉及到词语相似度的计算。在很多情况下，直接计算词语的相似度比较困难，通常可以先计算词语的距离，然后再转换成词语的相似度。相似度的计算公式为：

$$Sim(W_1,W_2)=\frac{\alpha}{Dist(W_1,W_2)+\alpha} \qquad (3.7)$$

式中，$Dist(W_1,W_2)$ 是W_1和W_2的词语距离，α为一个可调参数，其含义为当相似度等于 0.5 时的词语距离值。

词语距离有两类常见的计算方法，一种是根据某种本体知识（Ontology）或分类体系（Taxonomy）来计算，一种是利用大规模的语料库进行统计。

根据本体知识（Ontology）或分类体系（Taxonomy）计算词语语义距离的方法，一般是利用一部同义词词典（Thesaurus）。一般同义词词典都是将所有的词组织在一棵或几棵树状的层次结构中。在一棵树状图中，任何两个结点之间有且只有一条路径。于是，这条路径的长度就可以作为这两个概念的语义距离的一种度量。图 3.14 展示了词语间的距离。

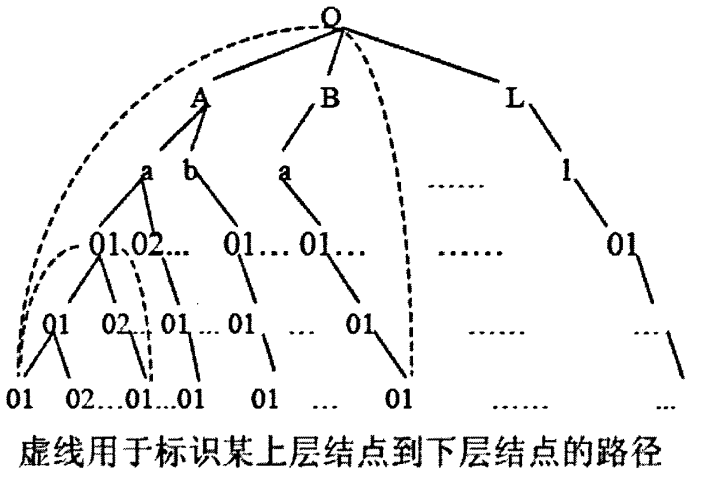

图 3.14 同义词词林语义分类树状图[36]

另一种词语相似度的计算方法是用大规模的语料来统计。可以利用词语的相关性来计算词语的相似度。事先选择一组特征词，然后计算这一组特征词与每一个词的相关性（一般用这组特征词在实际的大规模语料中在该词的上下文中出现的频率来度量），于是，对于每一个词都可以得到一个相关性的特征词向量，然后利用这些向量之间的相似度（一般用向量的夹角余弦来计算）作为这两个词的相似度。这种方法实质上仍旧是利用 VSM 模型去计算两个特征向量的相关度。与以往的情况不同的是，这两个特征向量一个是事先选定的特征词向量，另一个是由一个词语组成的特征向量（它的其它维实际上为 0）。这种做法的假设是，凡是语义相近的词，他们的上下文也应该相似。[37]研究了如何利用词语的相关性来计算词语的相似度。

本文对同义词比较的应用采用了上述第一种方法基于同义词词典来获取两个词语间的路径长度。这种方法依赖于所选用的词典。

目前作为计算词语间关系的词典有很多种。著名的有 WordNet[38]、中文知网（HowNet）[39]和《同义词词林》[40]。

WordNet 是一部在线词典数据库系统，是普林斯顿大学认知科学实验室的 Miller, Beckwith 等人自 1985 年起开发的，并一直不断地在进行改进、发展，目前发布的 windows 下的最新版本是 2.1 版本。是在当前基于人类词汇记忆的心理语言学理论推动下产生的，是一部由心理语言学家和计算机科学家共同努力下创建的基于英文的词汇语义网络系统。在 WordNet 中包含了众多的词语间的关

系，如同义关系、反义关系、上下位关系等。对于 WordNet 中词语的相似度计算的方法有 Hirst-St-Onge 方法、Leacok-Chodorow 方法、Resnik 方法等，这些方法在[41][42]中要详细介绍。

中文知网（HowNet）是一个以汉语和英语的词语所代表的概念为描述对象，以揭示概念与概念之间以及概念所具有的属性之间的关系为基本内容的常识知识库。知网中含有丰富的词汇语义知识和世界知识，为自然语言处理和机器翻译等方面的研究提供了宝贵的资源。

《同义词词林》是梅家驹等人于 1983 年编纂而成，初衷是希望提供较多的同义词语，对创作和翻译工作有所帮助。这本词典中不仅包括了一个词语的同义词，也包含了一定数量的同类词，即广义的相关词。《同义词词林》完全可以作为语义词典用到自然语言处理任务中。同时，《同义词词林》与 WordNet 的格式有若干相似之处，即都是用一个同义词集合来表示一个意思，所以可以对 WordNet 中的各种语义度量方法进行改造用于《同义词词林》。然而，由于著作时间较为久远，且之后没有更新，为了对其加以更新利用，哈尔滨工业大学信息检索实验室利用众多词语相关资源，并投入大量的人力和物力，完成了一部具有汉语大词表的《哈工大信息检索研究室同义词词林扩展版》，以下简称《扩展版》。

由于 WordNet 主要应用于英文词汇的关系计算，而知网在体系结构上并非一般词典的树状结构，而是真正的网状结构且包含的关系众多、使用复杂。与此相比，《同义词词林》在结构上与 WordNet 近似，相对简单而且适用于 WordNet 的一些相似度计算方法，所以本文选择《扩展版》作为词语关系计算的词典。在算法上使用基于词语路径长度的方法。

原始的《同义词词林》采用三层编码，《扩展版》在此基础上采用了五层编码，即大类、中类、小类、词群和原子词群。具体标记参见表 3.2 所示。

表 3.2 哈工大扩展版编码规则表

编码位	1	2	3	4	5	6	7	8
符号举例	D	a	1	5	B	0	2	=\#\@
符号性质	大类	中类	小类		词群	原子词群		
级别	第一级	第二级	第三级		第四级	第五级		

大类用大写英文字母表示，中类用小写英文字母表示，小类用二位十进制

整数表示。例如：“Ae 07 农民 牧民 渔民”,“Ae 07”是编码，“农民 牧民 渔民”是该类的标题。标题是由一个或者多个第四层的“段首（即每个段的第一个词）”组成。新增的第四级和第五级的编码与原有的三级编码和并构成一个完整的编码，唯一的代表词典中的出现的词语。如：

Ba01A02= 物质 质 素

Cb02A01= 东南西北 四方

对于第五级的结果还分为三类，即第八位的标记。表中的编码位是按照从左到右的顺序排列。第八位的标记有 3 种，分别是“=”、“#”、“@”，“=”代表“相等”、“同义”。末尾的“#”代表“不等”、“同类”，属于相关词语。末尾的“@”代表“自我封闭”、“独立”，它在词典中既没有同义词，也没有相关词。

《扩展版》数据词典分数据文件和索引文件。索引文件中任一条记录的格式如下：

lexicalName synsetNumber <synsetOffset>

数据文件中任一条记录的格式如下：

synsetOffset wordNumber <word> <synsetOffset pointerSymbol >

其中<>表示可以为有限多项，各个字段的含义如表 3.3 所示：

表 3.3 《扩展版》的文件格式说明

数据文件格式		索引文件格式	
字段名	含义	字段名	含义
synset0ffset	同义词集合编号，长度为 8 的字符串	lexicalName	词语名称
wordNumber	集合中单词的个数，用两位十六进制整数表示	synsetNumber	包含该词语的同义词集合的个数
word	各个词语名称	synsetOffset	包含该词语的同义词集合的编号
ptr	指针，包括 pointer_symbol，synset_offset		
synset_offset	目标集合在相应词性文件中的编号		
pointer_symbol	指针符号		

在数据文件格式中，pointer_symbol 的取值通常为上位关系$和下位关系~两种符号，举例如图 3.15 和图 3.16 所示。

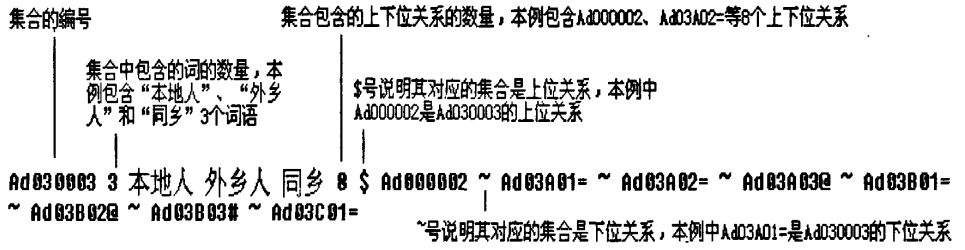

图 3.15 数据文件格式举例

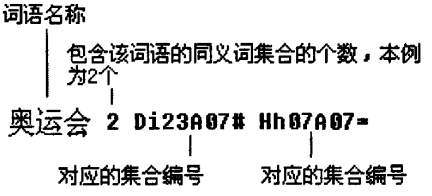

图 3.16 索引文件格式举例

基于《扩展版》的词语编码规则以及文件存储格式，本文给出使用《扩展版》进行相似度计算的接口定义。

```
/// <summary>
/// 上下位关系类
/// </summary>
public class Relation
{
    string relationSymbol;  //上下位标识
    string synSetId;  //集合编号
}
/// <summary>
/// 索引记录结构类
/// </summary>
public class SynSetIndex
{
    string word;  //索引的词语
    int synSetNum;  //对应的集合个数
```

```
        string[] synSets;  //对应的集合编号数组
    }
    /// <summary>
    /// 数据记录结构类
    /// </summary>
    public class SynSetData·
    {
        string synSetId;  //集合编号
        int wordNum;  //集合的词语数
        string[] words;  //集合的词语数组
        int relationNum;  //集合的上下位关系数
        Relation[] relations;  //集合的上下位关系数组
    }
```

根据图 3.14 所示，每一个词语都分别位于不同的叶子结点。因此对两个词语计算距离的算法如下：

1. 对待比较的两个词 A 和 B 分别定义一个距离变量 a 和 b, a 和 b 的初始值位 0。判断 A 和 B 所属的集合是否属于同一个集合，即同一叶子节点。如果是，则距离返回 a＋b 的值；否则，a，b 各增 1，转向 2。
2. 分别取 A 和 B 所属集合的上位集合，判断是否为同一集合。如果是，则距离返回 a＋b 的值；否则，a，b 各增 1，转向 3。
3. 判断 a，b 是否等于 5（由于同义词词林是五层编码）。如果是，则直接返回 a＋b；否则，转向 2。

值得注意的是，有些词语会属于多个集合。对于这样的词要对其所属的集合分别按照上述算法计算一组词语距离值。在此，我们采用乐观地方法选用距离值最小的为最终计算相似度使用。

对计算获得的词语的距离按照公式（3.7）计算其相似度，当相似度值大于某一阈值时，认为这两个词语为同义词。

3.3.3.2 抽取算法设计

查询表单入口的主要元素标签有 input 标签，其中包括类型为 text、button、

reset 和 submit 等。另外还有 select、radio 和 checkbox 等标签。然而表单上的这些标签本身不具备语义，通常是有文本与之对应来描述该属性的意义。我们把这些标签和与之对应的文本信息组合称为查询表单的属性，用来与本体域属性进行匹配。（对于表单属性的定义已经在 3.3.2.2 节中说明）因此，处理表单的第一步就是要将表单中的标签元素与文本对应起来使之代表一种属性信息。对于大多数表单来说，通常一个标签与对应的一条文本就可以代表一个属性。但对于有些表单，会出现多个标签表示同一属性，例如日期属性的年、月、日分别由三个 select 标签表示。这样就出现 1：m 的匹配方式。经过对大量表单的观察，这种情况在现实中出现的几率并不是很大，通常一些国外的电子商务网站有类似的情况。而且这种情况也根据表单查询主题的不同而不同，例如查询航班或酒店的表单可能会出现由三个标签表示的日期的这种情况，而对某些知识性内容的查询，如对药品功效的查询，则不会或很少出现这种情况。由此可见，1：1 的匹配方式在方法上实现简单，而且通常对大多表单是有效的。鉴于此，本文设计的模式抽取方法主要是针对 1：1 的匹配方式的。

目前对于从深度网查询入口表单中提取模式的方法普遍采用的都是基于网页标签和文本信息的位置关系建立起一些推论或者人工预先从大量网页中总结出来一些常用的位置关系的模板。例如，人们认为最接近标签的文本就应该是它代表的意义，但是在一些情况下这种假设是不准确的。类似的情况有，有的表单入口中的标签元素有一些说明性的信息，它们同样是距标签很近的文本，但通常它们不能直接代表标签的意义。此外，有的距离某个标签元素很近的文本信息可能代表其它标签元素的意义。对于图 3.13 所示的表单，药品名称文本对应的标签元素是第一个<input type=text>，而这个标签距药品分类文本有同样的距离，然而药品分类文本实际上是代表右边的 select 标签的意义。

由此可以看出，单纯地从位置和距离等信息得出的结论误差很大，需要有一组潜在的规则来处理位置信息。通过对网页的直观分析，可以得出以下规律：

1．同一个单元格内的元素相关的可能性比在不同单元格内的元素大；

2．在同一行内的元素相关的可能性一般比在不同行内的元素大；

3．子表内的元素之间相关的可能性比和不在这个子表中元素之间的相关性大。

4．对于标签元素四周都存在文本信息的情况，左边和上边的文本信息相关可能性大。

5．对于 radio button 和 checkbox 标签元素，对应的文本通常在标签的右边。

总结了上述的规律后，下一步要做的就是获取表单中各个元素的位置信息。

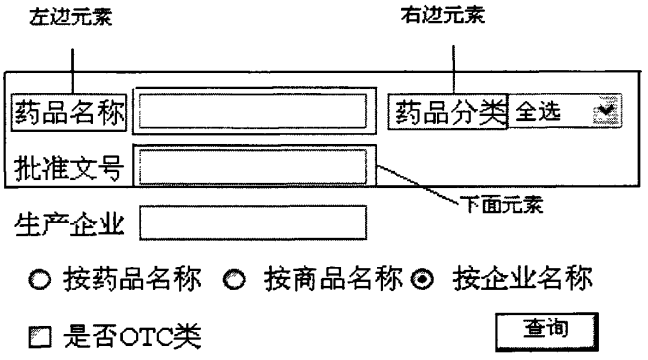

图 3.17 查询入口表单的区域划分

Manuel Álvarez等人在[43]中给出了一种计算元素间扁平距离的方法，描述如下：

在文本元素 T 和表单标签元素 F 之间的扁平距离如下计算：

1．获取包含 T 的矩形区域和包含 F 的矩形区域之间的坐标，如果 T 是在 HTML 的 table 单元格内，并且它是里面唯一的文本，则将该单元格的坐标赋值给 T。

2．获取的两个矩形间的最小距离。距离不是以像素计算，而是较粗粒度的单元记录（取网页中一个字符的大小）。

3．获取连接两个矩形的最短线的角度。此角大约为 $\pi/4$ 的倍数。

该算法对表单上的每个标签元素计算出一组相关的候选文本信息（这些文本信息通常分布在标签元素的四周）。然后对得到的候选文本信息采用下面的规则进行筛选：

1.将所有与 F 有较短距离 d 的文本加入 list(list 为 F 对应的文本元素列表)。

2.将文本（与 F 距离小于 k*d）加入按距离排序的 list（k 是可配置的因子通常设置为 5）。这步丢弃了那些与 F 较远的文本。

3. 有相同距离的文本则根据它们的角度排序。按角度（通常是$\pi/2$的倍数）由大到小排序。最终使左边的优先权高于右边，顶部高于底部。

根据对Manuel Álvarez的算法的简单描述发现该算法在计算距离时完全基于表单元素间的扁平位置关系，没有对网页HTML源代码的层次关系做分析，同时缺乏对语义的判定。因此，在此基础上，本文所设计的Deep Search Crawler改进了该算法，加入了对网页源代码的分析和语义要素。加入语义要素主要是将表单属性的文本信息与本体域属性名及别名进行语义上的匹配。

对于网页表单来说，尽管不同的开发者设计的页面风格迥然不同。但对于大多数页面来说，其中的标签元素通常所具有的某些属性会在一定程度上代表了该标签的意义。H. He 等人在[44]中，对这一规律有相关描述。例如，当要求查询表单时输入名称条件时，许多的网页表单中会有类似这样的标签元素。

```
<input type="text" size="16" name="med_name">
```

可见，name属性从直观上就可以断定该标签的意义是药品名称。此外，在通过对大量表单的观察，可以得出一些经验性的结论：

1. 对于文本信息为类别、分类等文字，它们所对应的标签元素通常是 select 下拉框标签，其中 select 标签的 name 属性可能会包含诸如“class”，“cls”之类。

2. 对于文本信息为名称、关键字等文字，它们所对应的标签元素通常是 text 输入框标签，其中 text 的 name 属性经常是“name”，“key”等等。

为此，定义一个集合$S=\{Z_1,Z_2,...,Z_n\}$。S中的每一个Z_i用一个二元组表示$Z_i=\{T,E\}$。T表示文本信息的词语，E表示标签元素。例如，T包含“类别”，E包含<select>。这样就建立起来一个文本与标签的对应信息{“类别”,<select>}。这个集合 S 是通过对一定数量的表单进行学习提取普遍规则来获得的经验集。

根据分析，本系统改进后的算法如下：

1. 首先根据Manuel Álvarez的算法计算标签元素和文本信息的距离，对每一个标签元素获取一组候选文本信息，并对文本信息按先距离后角度的顺序排序，选取距离最短的几个文本。

2. 对表单的HTML代码利用传统的DOM树的形式进行表示，分别对标签的候选文本信息对应的DOM结点同标签的DOM结点计算层次距离，匹配距离最近的文本。

3. 若上一步匹配不成功，通常为标签包含的最近的文本仍不只一个，则根

据文本信息中的词语按照预先定义的经验集合S进行匹配，哪个文本中的词语匹配标签元素，则将它们关联。

4．若上一步匹配不成功，则取候选列表的第一个文本作为匹配结果。

5．最后，将匹配的文本与本体域中的各属性的属性名及别名进行匹配，若不成功，则按照3.3.3.1节的算法进行语义上的判别，以期获取同义或同类的关系。

在获取元素间层次距离信息时，需要对页面进行解析。我们采用的是htmlparser 在.NET 下的版本 HTMLParser.NET 组件，可以很方便地生成页面的DOM 树，并可以获取文本信息节点。另外在计算距离时，采用 mshtml 组件，动态加载页面来获取标签元素的坐标位置。在生成最终页面属性的时候，首先将表单的属性与本体域的属性进行匹配，对表单上的某一属性若匹配不成功，则根据 3.3.3.1 节描述的算法进行同义词的匹配，若匹配不成功，则该属性与本体域无对应的匹配关系。此外，对于每个表单属性最终的候选查询条件，我们一般在设定本体域的时候就对相应的本体域属性设置一组查询条件。若表单属性匹配此本体属性，则选择该本体属性的查询条件。这通常用于 input 的 text 标签，对于 select、checkbox 和 radio 标签来说，它们的查询条件就是其标签中的值。

第四章 深度网爬虫的设计与实现

本文设计了一个深度网爬虫 Deep Search Crawler。爬虫旨在爬行主题页面的同时更多更有效地获取 Web 上丰富的深度网资源。爬虫主要实现两大功能模块，一是主题爬虫(Focused Crawler)，主要实现传统主题爬虫功能用来爬行一般的网页；二是深度网爬虫(Deep Web Crawler)，实现深度网爬行，主要对表单进行解析，抽取入口模式，并在此基础上通过 post 方法获得查询结果页面，进而爬行结果页面的链接。该爬虫除爬行传统的网页的同时，还爬行深度网页面，以获得更多的一般爬虫无法爬取的信息。并且采用一种在线关键词学习的方法来获取并调整关键词以判别相关度。

第一节 设计思想

本文的设计采用了模块化的思想，对系统整体功能做了明确的划分，以做到各司其职，分块处理。系统框架图如下：

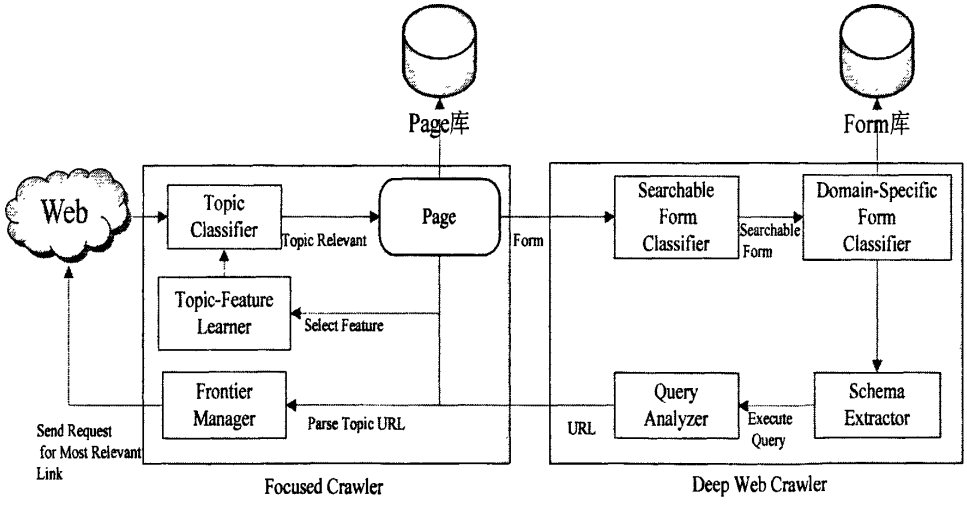

图 4.1 整体框架图

如图 4.1 所示，系统分为主题爬虫 Focused Crawler 和深度网爬虫 Deep Web

Crawler 两大模块。

主题爬虫 Focused Crawler 主要包括主题分类器 Topic Classifier、主题特征学习器 Topic-Feature Learner 和 URL 队列管理器 Frontier Manager。主题爬虫模块的工作模式为：

1. 初始化 Topic Classifier 和 Frontier Manager。主题分类器主要对爬行的网页计算主题相关度。主题相关度按照公式（3.3）进行计算。主题相关度的计算需要主题向量的支持，可以预先给定主题向量或通过一组样本学习获得，爬虫首先以预先给定的一组页面集合作为样本，逐一进行分词生成页面对应的词集并根据词的权重排序，以权重大于阈值的一组词作为主题特征词。生成的主题特征词向量表示如下：

$V=\{t_1,w_1;t_2,w_2;...;t_n,w_n\}$，其中，$t_i$ 表示一个特征词，w_i 表示该特征词的权值。

爬虫以主题特征词向量 V 为指导进行爬行。学习特征词的算法已在 3.3.1 节详细介绍过。对主题分类器的初始化即对其选定主题向量。URL 管理器用于维护一组 URL 队列，管理器的初始化就是对其选定一组种子 URL 站点。种子站点的选定一般是由人为给定。

2. 从 URL 管理器中提取排在首位的 URL，向服务器发送 Request 请求。服务器返回的页面提交给主题分类器，并由主题分类器计算其相关度。如相关度大于某一预先设定的阈值，则认为页面是主题相关，存入 Page 库。对主题相关的页面要解析其页面内的链接并对链接预测一个相关度值，然后将解析出的链接存入 URL 管理器。URL 管理器会自动将这些链接按相关度由高到低的顺序排序。

3. 对于主题相关的页面要提交主题特征学习器。主题特征学习器主要用于对相关的页面进行特征词的学习，具体的实现方法见 3.3.1 节。

4. 与传统方法的爬虫不同。本文所设计的主题爬虫对相关的页面还要进行表单的解析。对提取出的表单将作为输出结果提交给深度网爬虫 Deep Web Crawler 进一步处理。

深度网爬虫 Deep Web Crawler 主要由可搜索表单分类器 Searchable Form Classifier、主题表单分类器 Domain-Specific Form Classifier、模式抽取器 Schema Extractor 和查询分析器 Query Analyzer 组成。其工作模式为：

1. 对主题爬虫输出的表单进行处理。通过可搜索表单分类器过滤出不可搜索的表单形式，这主要包括登录表单、电子邮件表单、发表留言或跟帖形式的

表单和投票或订阅等形式的表单。

2.对提取的可搜索的表单提交主题表单分类器进一步过滤。提取与主题相关的表单。在这一步首先要定义本体域。本体域的定义用于过滤与查找领域不相关的表单，具体方法见3.3.2节。

3.对可搜索的主题表单提交模式抽取器抽取出它的查询模式。这一步的主要工作是将表单中的各个标签元素与表单中的文本元素尽可能地进行匹配以结合为表单的属性，具体方法见3.3.3节。实际上，在进行深度网爬行时，第2步与第3步是结合在一起进行的。然而，为了便于更清晰地介绍和划分功能，才将其分成两步说明。

4.利用查询分析器对抽取模式后的表单生成不同的查询提交到服务器，然后对服务器返回的结构进行解析。将结果中的链接提交到主题爬虫的URL管理器进行处理。

对整个系统实现的总体流程图如图4.2所示。从流程图中可以看出，在系统开始爬行的时候，首先要对系统进行必要的系统配置（Configuration），这包括设定种子站点、选择主题词集、配置爬行深度、使用的线程数、相关度计算的阈值及输出目录等信息。系统配置的种子站点地址存入URL管理器。对于种子站点的初始相关度预测值设为1，即认为种子站点是绝对相关的。在爬行时，首先从URL管理器中获得一个URL地址，然后向站点服务器发送请求。在服务器返回页面后，对返回的页面进行相关度的计算，将相关度大于阈值的网页进行保存。同时提取页面上的链接地址，对其进行相关度预测。若预测的相关度大于事先设定的阈值，则将该地址存入URL管理器。注意，在对已抓取的网页进行相关度计算时所使用的阈值与对页面链接预测时使用的阈值不同。通常情况下，前者所使用的阈值要高于后者，这样可以保证抓取页面的主题相关性。此外，对于页面中预测相关度低于阈值的链接采取的策略是并不马上丢弃，而是存入URL管理器继续对其爬行同时记录一个爬行深度，只有当爬行深度达到预设的值且仍然主题无关才将其丢弃。当两个主题相关页面之间通过无关页面连接时，这样做可以有效地爬行到更多的相关页面。

对于抓取到的页面如果主题相关，则搜索页面是否包含查询表单的入口。在搜索表单时，要排除如登录表单、电子邮件表单和投票订阅之类的表单等等。如果包含查询性质的表单，则对其进行相关度的判别和模式的抽取，并根据预设的查询候选条件进行查询。对查询返回的结果页面进行解析，以获得页面上

的地址链接。这些链接将存入 URL 管理器以继续爬行。

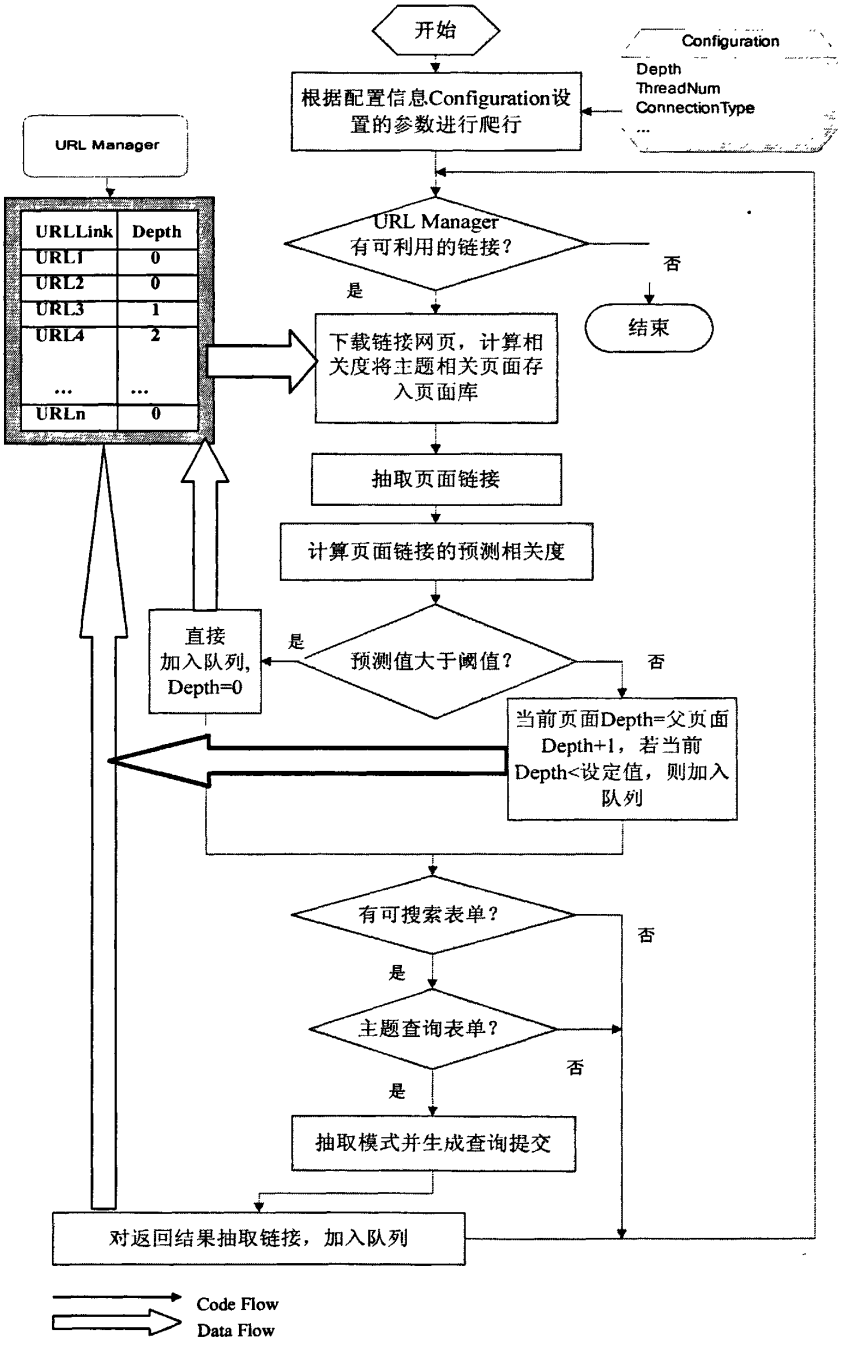

图 4.2 系统整体流程图

第二节 系统实现

本节主要描述了 Deep Search Crawler 爬虫的实现。软件开发环境为 VS2005.NET，开发语言：C#。数据库：SQL Server2000。

本节将主要介绍系统实现的各个部分和类的定义。以下将分别从 URL 管理器、多线程爬行、深度网表单解析和相关度计算四部分来说明系统在具体实现中的处理方法。图 4.3 为系统运行主界面。

图 4.3 主题爬虫程序主界面

4.2.1 URL 管理器

URL 管理器模块主要是维护一组 URL 队列，并对 URL 按照主题相关度的预测值进行排序，按照“best first”的调度策略，将预测值最高的 URL 分配给爬行器。URL 管理器所维护的队列有等待队列、运行队列、完成队列和错误队列。由于基于内存的队列维护无法解决实际 URL 数量过多的问题，本系统采用 SQL Server 数据库表来模拟队列。图 4.4 为存储 URL 队列的 SQL Server 表。

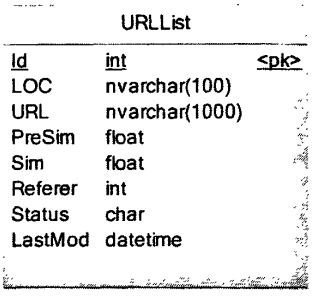

图 4.4 URLlinkList 表字段

其中，Id 为数据库自动增 1 的标识，作为主键应用；LOC 为 URL 链接的永久地址，存储链接的域名；URL 为地址链接；PreSim 为链接的预测相关度；Sim 为链接的相关度；Referer 为链接的父结点网页的数量，即有多少链接指向该页面；Status 为链接的状态，一般为等待“W”、下载中“R”、完毕“C”和错误“E”；LastMod 为最近一次爬行的时间。

URL 管理器的 URL 主要来源于以下两个方面：

1．初始的种子集。

2．URL 提取器从下载的页面中提取的新 URL。

对于种子集中的 URL，一般是由专业领域人员人工设定，或是通过通用搜索引擎获得主题网站的列表，如 Yahoo！分类目录。对于种子集中的 URL 的主题相关度的预测值设定为 1。种子集中的站点具有最高优先级。对于从已下载的页面中提取的新的 URL，则根据主题判别模块的相关度预测计算得出。当相关度高于某一阈值，则加入等待队列。否则，对其设定无关页面的爬行深度。

在编程实现中，PreSim 存储的内容为该链接父节点网页的相关度平方的累加；Sim 为该链接的相关度预测值。在获取一个新的链接 URL 后，查找数据库的表中，是否已有该条记录。若没有，则直接插入，并将 PreSim 字段赋值为该 URL 父节点网页的相关度的平方。Sim 字段则是根据公式（3.5）计算出的相关度预测值。若表中已存在该 URL 记录，则将此次获得该 URL 的父节点页面的相关度平方累加到 PreSim 字段并重新按公式(3.5)计算 Sim 字段的相似度值。对于已经爬行过的网页，Sim 字段更改为该网页的实际相关度。

4.2.2 多线程爬行

多线程是指程序在同一时刻可以运行多个执行流程。这使得计算机看起来就象能够同时执行一个以上的操作任务。对于爬虫来说，要完成爬行任务必须对每个 URL 链接的页面进行下载，这种下载量是相当大的。爬虫需要向 Web 站点服务器发送请求，然后接收这些网页。对于 Web 的请求不同于在本地运行，爬虫在发出请求后，必须等待服务器的响应，这是一个异步的过程。

如果单线程爬行的话，对于爬行的所有链接都要有一个等待过程，因此总的等待时间就是对所有链接发送请求的等待时间之和。这样的效率显然不能满足对大量 URL 下载任务的要求。

多线程的引入而能够有效地减少总下载的时间。使用多线程爬行，对每个 URL 请求的等待时间可以有效地叠加，这样就大大减少了总的等待时间，提高爬行效率。然而，由于每条线程运行时都要占用一定的资源，所以在使用多线程爬行时不可无限制地增加线程数，而是要寻找一个合理的平衡点，使得既能高效地爬行网页，又不过分占用资源。

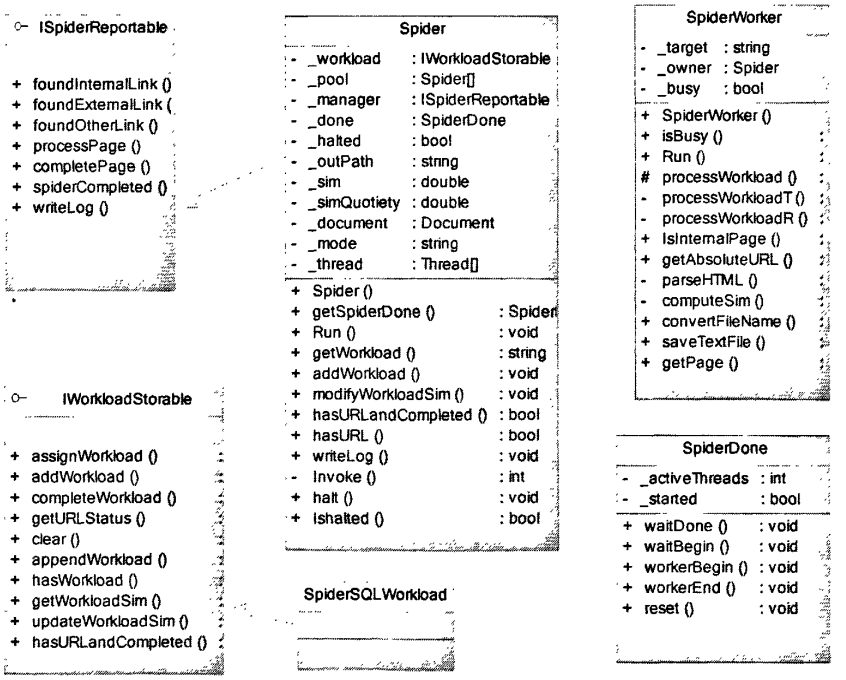

图 4.5 多线程爬虫的类定义

图 4.5 是使用多线程技术的爬虫的类结构定义，主要分为管理者类 Spider 和工作者类 SpiderWorker。管理者类作为程序主线程以及添加及获取链接所使用。工作者类作为工作线程的实体，负责具体的抓取页面及分析工作。类结构定义参照[45]。SpiderSQLWorkload 类完全实现 IWorkloadStorable 接口。另外，采用多线程爬行，需要判别何时工作线程全部完成任务。SpiderDone 类设计用于判别是否所有线程都已经工作完毕，其中包含活动线程数用于记录当前正在执行的线程的数量，当它为 0 时，系统认为所有线程工作完毕。

4.2.3 深度网表单解析

深度网表单解析的主要功能就是将表单查询入口的各组查询条件（schema）提取出来并自动生成查询条件进行提交查询。这一部分主要分为抽取模式和模拟查询两部分。

4.2.3.1 抽取模式

在对表单查询入口进行模拟查询前的关键问题就是要获取表单的各组查询条件。系统采用以表单中各元素的平面距离以及元素间的角度关系来匹配查询元素与哪一文本对应。由于对表单元素计算它们之间的平面距离需要获取各个元素的位置信息。此外，对表单进行查询时，需要模拟浏览器的事件来进行查询。因此，基于上述两点，在深度网入口抽取模块中使用 Microsoft IE Browser 组件进行动态加载页面以获得各个表单元素的位置。

抽取模式在实现上的编程思路是，首先对 Form 表单中的元素生成一棵 DOM 树。在 DOM 树的结点元素中，通过 MS IE Browser 组件获取其位置信息并将位置信息保存。我们定义了带位置信息的 DOM 树的存储结构如下：

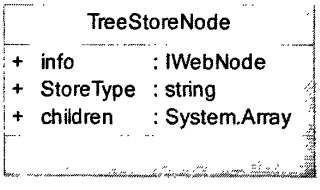

图 4.6 结点存储结构

TreeStoreNode 类定义了树的结点的结构，如图 4.6 所示。其中 children 为 ArrayList 类型的数组，存储该结点对应的子结点。StoreType 我 string 类型的变

量，用于标识该结点是否为叶子结点。因为在解析的 DOM 树中，实际上查询元素总是位于树层次的最底层，即叶子结点。因此，通过判断给结点是否为叶子结点间接上得出该结点是查询元素或者是文本。

TreeStoreNode 中最重要的一个属性是 info。它是一个接口类型，定义了各种类型的元素的公共属性及方法，如图 4.7 所示。其中对于表单中的各种类型的元素都实现此接口。

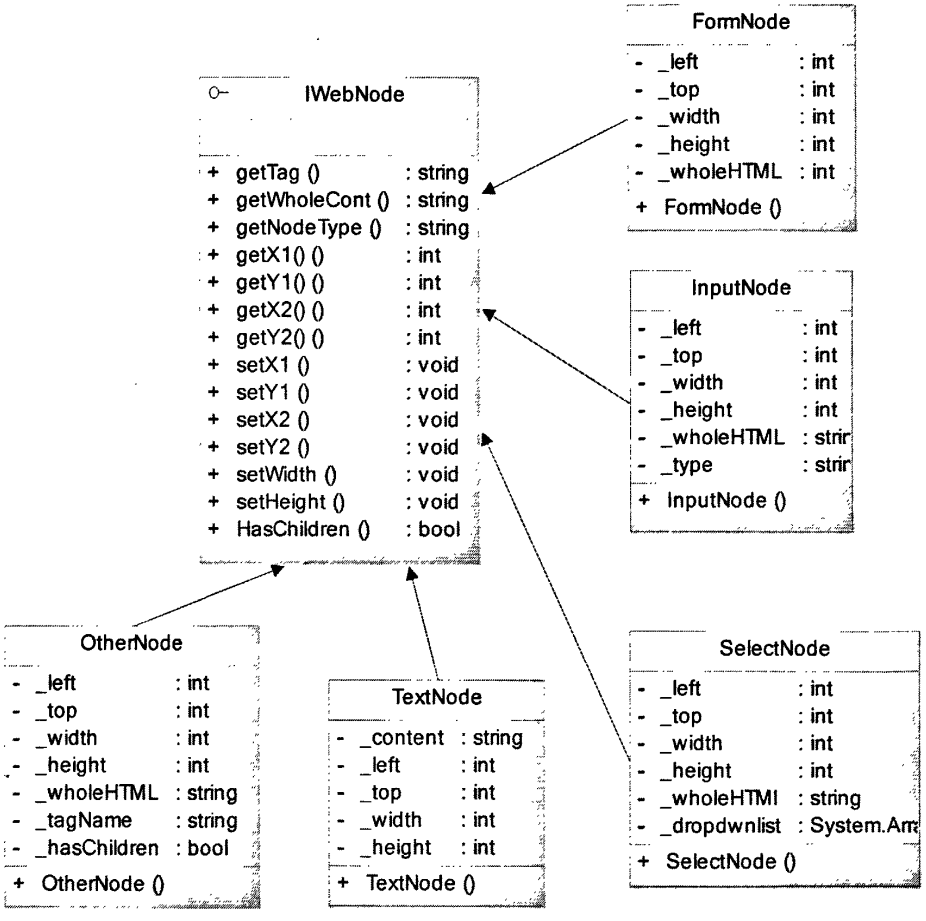

图 4.7 各标签元素的结点结构

FormElemType 类定义了查询元素的各种类型，包括 select、input button、input checkbox、input image、input file、input hidden、input pwd、input radio、input reset、input submit、input text、label 和 form。

Distance 类用于计算各元素间的平面距离和元素间的夹角。

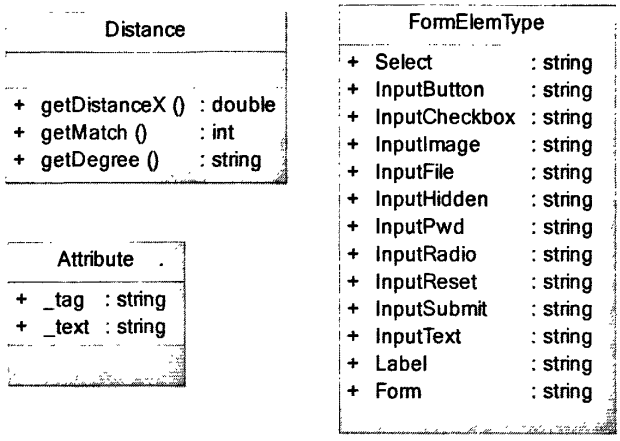

图 4.8 Distance、FormElemType、Attribute 类定义

将查询元素和与它最近的文本匹配组成一组属性，以 Attribute 类定义的结构进行存储。

4.2.3.2 模拟查询

模拟查询提交用于对解析后的表单生成查询条件并提交到服务器。对于上一步抽取模式获得一组 Attribute 类存储的属性，每一个属性由对应的文本和查询标签元素组成。在查询提交时，首先对属性的查询标签元素赋予一定的查询条件。赋值需要对表单的属性与本体域进行匹配，参见表 3.1。

对与本体域匹配成功的表单属性依次赋查询条件并提交。关键代码如下：

```
private void submit()
  {
   try
    {
      ArrayList attributes = getMatchAttribute();
      IHTMLDocument2 DOM = (mshtml.IHTMLDocument2)axWebBrowser1.Document;
      HTMLInputElement submit;
      for (int i = 0; i < attributes.Count; i++)
      {
       string type = "";
       HtmlElement he = getHtmlElement((DOMHTML.dist.Attribute)attributes[i], ref type);
```

```
switch (type)
 {
  case "text":
      {
       IHTMLInputTextElement input;
       //获得属性中tag元素的name
       string name = getName((DOMHTML.dist.Attribute)attributes[i]);
       //获得DOM查询元素
       input = (IHTMLInputTextElement)DOM.all.item(name, null);
       //根据对应属性赋查询条件
       input.value = getQuery(((DOMHTML.dist.Attribute)attributes[i]).Text);
       break;
      }
  case "radio": break;
  case "checkbox": break;
  case "select": break;
  case "submit":
       {
        string name = getName((DOMHTML.dist.Attribute)attributes[i]);
        //获得提交按钮对象
        submit = (HTMLInputElement)DOM.all.item(name, 0);
        break;
       }
   }
 }
 if (submit != null)
      submit.click();
}
catch (Exception ex)
{
   MessageBox.Show(ex.ToString());
```

```
        }
    }
```

Submit 函数展示了提交表单的全部过程，用于对解析出的表单进行查询提交。其中，getMatchAttribute()函数用于返回一个 ArrayList 型数组，存储与本体域匹配的表单属性集合。getHtmlElement()函数返回 HtmlElement 类型的属性，并返回该属性的 type 值，用于在循环中判断对属性做何操作。对各个 input text 标签赋条件后，执行提交按钮的 click 函数进行提交。

整体解析和提交的过程如图 4.9 所示。

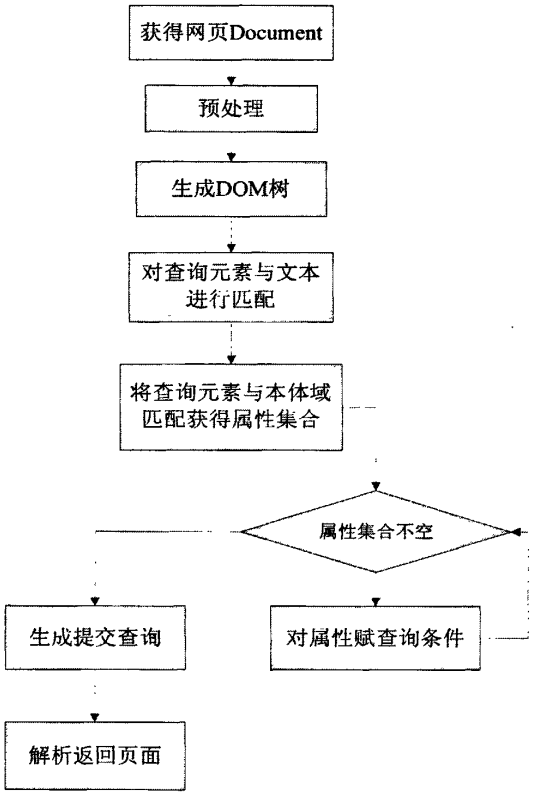

图 4.9 深度网入口处理流程

4.2.4 相关度计算

主题相关度判断需要使用专业词库，即构建主题向量。考虑词库中收集的

不同词汇对该主题的重要程度也不同，因此在构建词库时应对每个词汇设置权重。词库的构建可以采用人工收集的方式。此外，也可以通过对某一专题站点进行站内爬行，处理站内页面来获得词集，然后移除该词集中的常用词汇便得到一个专业词库。然而这种方法通常依赖所选择的种子站点是否专题明确，而且所获得的主题词库并不十分准确，通常要辅以人工参与的方法。

在本系统中采用学习的方法来获得主题向量。通过线下学习来获得初始向量，通过在线学习来调整主题向量的权值以及主题特征词的更新。在对页面进行解析时，使用了 HTMLParser。HTMLParser 是一个基于 Java 的开源的实时 web 页面抓取分析工具，它以其设计的简单性,分析实时 web 页面的高速性吸引了众多的开发者和使用者。本系统开发环境为 VS2005.NET,在系统中使用了 HTMLParser 的.NET 版本作为页面解析的主要工具。

在主题爬虫模块中，主要的相关度计算的类有三个，分别是 Text 类，Document 类和 Word 类。

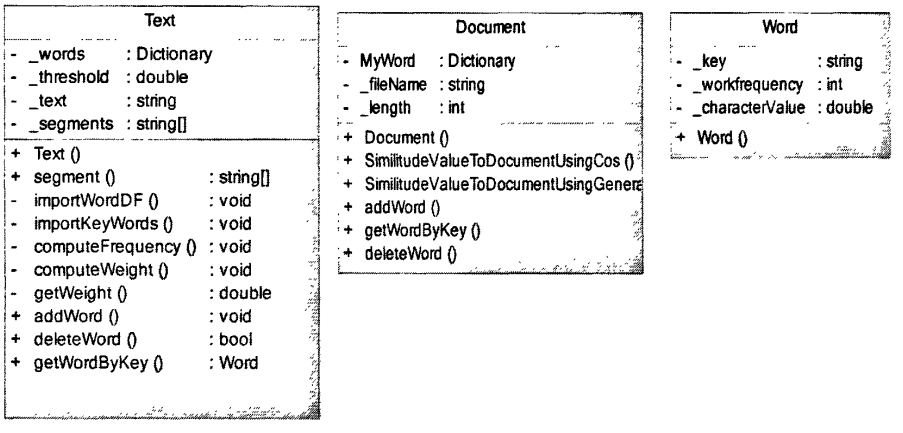

图 4.10 Text、Document、Word 类的定义

Text 类主要用于对给定页面的关键词学习。对每一目标页面的内容实例化为一个 Text 对象，并在调用构造函数时提取关键词。Text 构造函数如下：

```
public Text(string txt, double threshold)
{
    _threshold = threshold;
    KeyWord.N += 1;
```

```
        _text = txt;
        _segments = segment(_text);
        computeFrequency();
        importWordDF();
        computeWeight();
        importKeyWords();
    }
```

可以看出，在调用构造函数时已经调用 computeFrequency(),importWordDF(),computeWeight(),importKeyWords() 四个私有函数。computeFrequency()是对分词后的文本计算每个词的词频。importWordDF()是将特征词词典的数据导入 WordDF，WordDF[str]中存储的是 DF,即出现特征词的文档数目。这里 WordDF 是定义的关键词词典，它主要存储每个特征词的 DF 值。computeWeight()则是根据 TF*IDF 方法计算的关键词的权值，并对计算后的权值做归一化处理。最终，利用 importKeyWords()则是将权值大于 threshold 的特征词加入到关键词词典。

对于样本页面集合中的任一页面都要进行构造 Text 实例以进行关键词提取。这是一个迭代的过程，对每一页面提取关键词后，都要对关键词词典进行权值的调整以及关键词的更新。当对页面集合中的每一个页面都提取关键词后，则形成最终的初始主题向量。

Document 类主要用于爬行页面时的相关度计算。

主题相关度计算过程如下：

1.对页面的正文进行分词，去掉停用词，保留关键词，然后按照关键词在文章中出现的频率对关键词加权处理;

2.根据设定主题中的特征向量对得到的页面关键词进行调整。

3.按照公式 (3.3)计算出页面 D 的主题相似度 Sim(D)。

4.根据 Sim(D)值的大小和阀值 d 进行比较，如果，Sim(D)大于等于 d，则表示页面与主题相关，保留到数据库中;否则判为不相关，丢弃该页面。

Word 类主要作为关键词及词频和权重的存储数据结构。

在对页面中的各个链接预测相关度时，系统采用了基于父结点页面的主题相关度预测。一般对链接进行主题相关度预测时，对锚的文本信息计算相关度是必不可少的。由于锚文本可能本身并不具有主题意义，例如，锚文本为“详

细查看”。这时，需要引入扩展锚文本的概念，即锚文本以及锚周围的文本信息。实际操作时，通常取锚的上下文的若干字符来代表锚链接的主题意义。

对于扩展锚文本的信息采用分词、提取关键词向量按（3.3）公式计算相似度。

第三节 实验结果分析

4.3.1 实验结果

在实验中，分别选则九州通医药网 http://www.jzteyao.com.cn 和中国药网 http://www.pharmnet.com.cn 作为种子站点进行站内爬行。爬行的结果数据如表 4.1 所示。

表 4.1 比较结果

Web 站点	Form 数量	去重后的 Form 数量	静态链接	爬行总链接
http://www.jzteyao.com.cn	993	2	11212	21049
http://www.pharmnet.com.cn/	3393	15	35300	43151

由表 4.1 可以看到，在爬行九州通医药网时，爬虫获取的深度网查询入口表单数为 993 个。经过对表单的去重，最后得到 2 个有效表单。在不对表单进行查询时，所爬行的静态链接数为 11212。通过对表单利用表 4.1 定义的本体域进行查询后，对返回的链接进行爬行层数为 2 层的再爬行。最终，爬行到的链接总数为 21049。在这里，由于对一般的表单入口查询后，通常获取一组结果的列表。对每一个结果会有一个详细信息页，一般不会超过 2 层，所以对表单查询后返回的链接选择 2 层的爬行。对于中国药网的爬行类似。

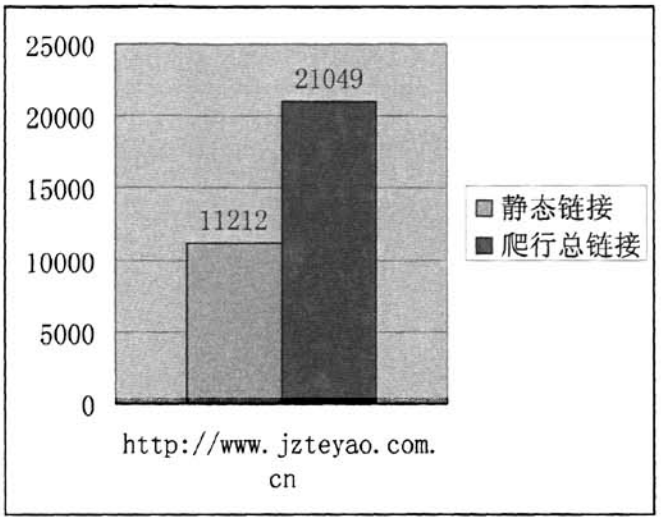

图 4.11 九州通医药网爬行链接图

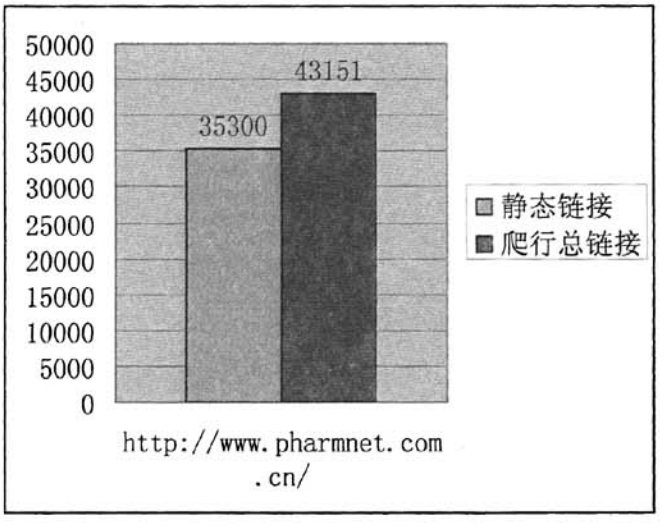

图 4.12 中国医药网爬行链接图

4.3.2 分析

由图 4.11 和图 4.12 所示可以看出，对九州通网的爬行效果要好于中国药网。

对于九州通网进行静态链接爬行和深度网爬行的对比发现。深度网爬行获取的链接在普通的静态链接爬行中是不存在的。例如，http://ec.jzteyao.com/view/jsp/gmit/jzt/buying/OrgMerchandiseTree/Merchview.jsp?currMerchandiseOID=62241 。这是通过深度网爬行获得的页面链接。然而，对于该链接的目录部分 http://ec.jzteyao.com/view/jsp/gmit/jzt/ buying/OrgMerchandiseTree/在只爬行静态页面时，URL 队列并不包含以次目录为前缀的页面。因此，对深度网的爬行可以更大限度的获取有用信息。

然而对中国药网来说，它的网站大部分是扁平化的，所以大部分查询表单后的结果页面可以通过静态链接来访问到。

第五章 总结与展望

第一节 系统总结

Web 信息量的急剧增长和门类领域的不断扩充使得通用搜索引擎越来越无法满足用户对某一主题的搜索要求。这在一定程度上激发了垂直搜索引擎的发展。主题爬虫作为垂直搜索引擎的关键组成部分，具有较高的理论和现实的研究价值。与此同时，对深度网的应用已经非常广泛的深入到人们的日常生活中，针对这方面的研究也是越来越多。由于深度网具有信息量大、数据质量高，因此对于深度网爬行技术的研究具有重大的现实意义。

本文对深度网爬行所面对的问题进行了概述。针对深度网入口的定位和入口模式的抽取分别提出了基于本体域的入口定位方法和基于网页标签距离及语义判别的入口模式的抽取方法。并且对主题爬虫的特征词采集提出了一种在线学习的特征词训练方法。实践证明，本文提出的深度网入口定位和模式抽取的方法可以较好地抽取与主题相关的表单的模式。同时，对主题特征词的在线学习也可获得区别主题能力较强的主题词集。

第二节 展望

本文设计的覆盖深度网的主题爬虫对大多数查询表单可以很好地抽取其入口模式。但是，有一些表单其内部的某些查询条件是通过脚本动态生成。另外，也有一些查询表单完全是通过脚本动态生成的，脚本通常为 Javascript。对于这类表单，本文所设计的爬虫尚无法进行访问。在未来的研究中，可以着重研究对动态脚本的解析，使之可以解析脚本生成的表单。此外，对于当爬行深度加大后，网页下载速度和处理能力都要受到限制。因此，对 Web 应用分布式爬行可以成为未来解决爬虫效率问题的重要方法。

参考文献

[1], L. Barbosa and J. Freire, Searching for Hidden-Web Databases, In Proc of WebDB, 2005, pages 309-321

[2], L. Barbosa and J. Freire, An Adaptive Crawler for Locating Hidden-Web Entry Points, In Proc of IW3C2, May 8-12,2007

[3], K.C.-C. Chang, B. He and Z. Zhang, Toward Large-Scale Integration: Building a MetaQuerier over Databases on the Web, In Proc of CIDR, 2005, pages 44-55

[4], S. Chakrabarti, M. van den Berg, B. Dom, Distributed hypertext resource discovery through examples, In Proc of the 25th VLDB Conference, Edinburgh, Scotland, Morgan-Kaufman, 1999, pages375-386

[5], M. Diligenti, F. M. Coetzee, S. Lawrence, C. L. Giles and M. Gori, Focused Crawling Using Context Graphs, In Proc of the 26th VLDB Conference, Cario, 2000,pages 527-534

[6], Taher H. Haveliwala, Topic Sensitive PageRank, In Proc of WWW 2002, Honolulu, Hawaii, USA, May 7-11,2002

[7], Google Base, http://base.google.com

[8], J. Xu and J. Callan, Effective retrieval with distributed collections, In Proc of SIGIR, 1998, pages112-120

[9], C. Yu, K.-L. Liu, W. Meng, Z. Wu, and N. Rishe, A methodology to retrieve text documents from multiple databases, TKDE, 2002, 14(6):1347-1361

[10], L. Barbosa and J. Freire, Siphoning Hidden-Web Data through Keyword-Based Interfaces, In Proc of SBBD, 2004, pages309-321

[11], S. Raghavan and H. Garcia-Molina, Crawling the Hidden Web, In Proc of VLDB, 2001, pages129-138

[12], Brightplanet's searchable databases directory, http://www.completeplanet.com

[13], W. Wu, C. Yu, A. Doan, and W. Meng, An Interactive Clustering-based Approach to Integrating Source Query interfaces on the Deep Web, In Proc of ACM SIGMOD, 2004, pages95-106

[14], G. Salton and C. Buckley, Term Weighting Approaches in Automatic Text Retrieval, Information Processing and Management, 1988, 24(5):513-523

[15] T. Joachims, A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Catergorization[C], In Proc of the 14th ICML, 1997, pages 143-151

[16], J. Cho, H. Garcia-Molina and L. Page, Efficient crawling through URL ordering, In Proc of the 7th International World Wide Web Conference, Brisbane, Australia, Elsevier Science, Apr,1998, pages 161-172

[17], P. DeBra, G. Houben, Y. Kornatzky, et al, Information Retrieval in Distributed Hypertexts[A], In Proc of the 4th RIAO Conference[C], New York: Computer assisted Information Retrieval, 1994, pages481-491

[18], H. Michael, J. Michal, M. S. Yoelle, The Shark Search Algorithm-An Application: Tailored Web Site Mapping[J], Computer Networks and ISDN Systems, 1998, 30:317-326

[19], F. Menczer and A. E. Monge, Scalable Web Search by Adaptive Online Agents: An InfoSpiders Case Study,1999, pages 323-347

[20], http://www.lib.berkeley.edu/TeachingLib/Guides/Internet/InvisibleWeb.html

[21], Ali I. El-Desouky, Ali A. Hesham, M. El-Charmrawy Sally, An Automatic Label Extraction Technique for Domain-Specific Hidden Web Crawling(LEHW), In Proc of IEEE, 2006

[22], K.C.-C. Chang, B. He, C. Li, M. Patel, and Z. Zhang, Structured databases on the web: Observations and implications, In Proc of ACM SIGMOD, Sept.2004, Record,33(3):61-70

[23], J. Madhavan, PA Bernstein, E. Rahm, Generic Schema Matching with Cupid, In Proc of the 27th VLBB Conference, Rome, 2001

[24], S. Melnik, H. Carcia-Molina, E. Rahm, Similarity Flooding: A Versatile Graph Matching Algorithm, In Proc of the 18th ICDE, San Jose, Feb,2002

[25], A. Doan, P. Domingos, A. Halevy, Reconciling schemas of disparate data sources: A machine-learning approach, In Proc of ICMD, SIGMOD, Santa Barbara, California, USA, ACE Press, 2001

[26], AH Doan, P. Domingos, A. Levy, Learning source descriptions for data integration, In Proc of WebDB, 2000, pages 81-92

[27], 俞士汶, 计算语言学概论[M], 北京: 商务印书馆, 2003

[28], Yang Yiming, P. O. Jan, A Comparative Study on Feature Selection in Text Categorization[C], In Proc of the 14th ICML, San Francisco, 1997, pages 412-420

[29], K. Aas, L. Eikvil, Text categorization: A Survey[Z], http://www.nlp.org.cn, 1999

[30], Manning, 统计自然语言处理基础[M], 苑春法译, 北京:电子工业出版社, 2005, 249-344

[31], D. Bergmark, C. Lagoze and A. Sbityakov, Focused Crawls, Tunneling, and Digital Libraries, In Proc of the 6th European Conference on Digital Libraries, Rome, Italy, Sept, 2002

[32], D. Gibson, Jon M. Kleinberg, P. Raghavan, Inferring Web Communities from link Topology, In Proc of the 9th ACM Conference on Hypertext and Hypermedia, 1998

[33], N. Eiron and K. S. McCurley, Analysis of Anchor Text for Web Search, In Proc of ACM SIGIR, Toronto, Canada, 2003

[34], R. Neches, T. Finin, Enabling technology for knowledge sharing, AI Magazine, Dec,1991, pages 36-56

[35], R. Studer, V. R. Benjamins, D. Fensel, Knowledge Engineering: Principles and Methods[J], Data and Knowledge Engineering, 1998, 25(1-2): 161-197

[36], 刘群, 李素建, 基于《知网》的词汇语义相似度计算, 中科院计算所, 2002

[37], 鲁松, 自然语言中词相关性知识无导获取和均衡分类器的构建, 中科院计算所, 2001

[38], http://wordnet.princeton.edu/

[39], http://www.keenage.com/

[40], 梅家驹, 竺一鸣, 高蕴琦等, 同义词词林, 上海:上海辞书出版社, 1983

[41], A. Budanitsky and G. Hitst, Semantic distance in WordNet: An experimental,

application-oriented evaluation of five measures, NAACL-2000, June,2001

[42], P. Resnik, Using information content to evaluate semantic similarity, In Proc of the 14th International Joint Conference on Artificial Intelligence, Montreal, Canada, 1995

[43], A. Manuel, R. Juan, P. Alberto, C. Fidel, B. Fernando, and C. Victor, DeepBot: A Focused Crawler for Accessing Hidden Web Content, In Proc of the 3rd DEECS, San Diego, California, USA, June 12,2007

[44], H. He, W. Meng, C. T. Yu, Z. Wu, Constructing Interface SCHEMAs for Search Interfaces of Web Databases. WISE 2005: 29-42

[45], J. Heaton, 网络机器人 Java 编程指南，童兆丰，李纯，刘润杰译，北京:电子工业出版社，2002，211-281

致　谢

首先衷心感谢我的导师邵秀丽教授。

就要离开生活了近三年的南开大学，往事历历在目。三年的研究生学习使我开阔了眼界、砥砺了心志，这与恩师的教诲和同窗的帮助是密不可分的。在近三年的学习生活中，邵老师给了我无微不至的关怀和帮助。本文的工作从研究方向的确定到论文的选题到最后的定稿都是在邵老师的悉心指导下完成的。此外,邵老师敏捷的思维、孜孜以求的进取精神给我留下了深刻的印象，开放的头脑、民主的学术作风、严肃认真而又不拘泥于形式的治学态度使我受益非浅。值此论文完成之际,谨向邵老师致以深深的敬意和由衷的感谢。

同时感谢实验室所有的同学，感谢他们给予我的关心、帮助和支持。还要感谢与我一起走过人生青春岁月的研究生同学;感谢培育了我三年的母校;感谢教过我课程、指导过我学习的无数老师;感谢帮助过我的所有人。最后，郑重感谢支持和鼓励我完成学业的父母和亲友。

由于自身知识的不足，文章必存在不少疏漏与缺陷敬请各位专家评委批评指正，在此表示衷心感谢!

个人简历、学术论文与研究成果

【基本信息】
姓名：陈磊
性别：男
出生年月日：1981 年 3 月 22 日
政治面貌：团员
【学习经历】
1999 年 9 月至 2003 年 7 月就读于天津工业大学计算机学院计算机科学与技术专业，获学士学位；
2005 年 9 月进入南开大学信息技术科学学院计算机科学与技术系，攻读计算机应用技术专业，获硕士学位，将于 2008 年 7 月毕业。
【所获奖励】
本科期间获得学校二，三等奖学金各一次
【参与项目】
天津正信集团 OA 办公系统
天津天士力之骄药业有限公司项目管理系统
天津医药集团医药商网