深度 web资源搜索关键技术

陈 鹏 刘烈宏

(北京航空航天大学 计算机学院, 北京 100191) (中国电子信息产业发展研究院, 北京 100846)

摘 要: 深度 web资源是指通过 web搜索等形式能够访问的网络数据库资源,由于它与静态网页存在着一些差异,传统的搜索引擎技术不能够很好的实现深度 web资源搜索. 研究了深度 web资源搜索中的若干关键技术,包括深度 web资源的自动发现和基于本体的深度 web数据抽取. 通过实验验证了所提出技术的可用性与高效性. 设计和实现了一种新的针对深度 web资源的搜索引擎系统,能够获取深度 web资源信息,并且利用这些信息抽取出结构化数据,从而为用户或者其他应用系统提供服务. 已经应用在国家重大工程项目"国家科技基础条件平台门户应用系统"中,并取得了很好的应用效果.

关 键 词: 搜索引擎; 信息抽取; 语义

中图分类号: TP 302. 1

文献标识码: A **文 章 编 号**: 1001-5965(2009)01-0013-05

Key technology of deep web search

Chen Peng

(School of Computer Science and Technology, Beijing University of Aeronautics and Astronautics, Beijing 100191, China)

Liu Liehong

(China Center of Information Industry Development, Beijing 100846, China)

Abstract: Deep web is web database accessed by the web search, it is different from surface web, so technology of traditional search engines is not applied to deep web research very well. Some key technology of deep web search was researched, including automatic discovery of deep web resource and ontology-based information extraction of deep web. Effectiveness of these technologies and relevant algorithms were verified through experiments. Based on these technologies and relevant algorithms, a deep web search system was implemented, it could access deep web resource and extract structured data for various applications. The system has been applied to national important project "national portal of science and technology infrastructure" and has achieved obvious effect.

Key words: search engines; information retrieval; semantics

由于 web技术的发展,已有的及新增的大量信息更多地采用动态的形式存在,而它们却没能被传统的搜索引擎、检索工具所检索,形成了大量的不可见的深层网络信息[1]. 这个概念是 Dr. Jill Ellsworth在 1994年首次以"Invisible Web"一词提出来的. 在 2001年, Price和 Sherman把它定义为 Deep Web,即为通过万维网可以获得的文本网页、文件或其它高质量、权威性的信息资源,但是由于技术限制或者特定选择的原因,使得通用搜索引擎不能够或不愿意把这些信息加入网页索引库中.

最近美国伊利诺斯州大学计算机科学系对深度 web被访问利用的情况所做的一份调查得到以下结论[1]:当前深度 web的规模为 307 000个站点, 450 000个数据库和 1 258 000个查询接口,在 2000～2004年间增长了 3～7倍;深度 web广泛

收稿日期: 2008-01-18

基金项目: 国家科学技术部基金资助项目(2005DKA63901)

作者简介: 陈 鹏(1979 -),男,江西南丰人,博士生, chenpeng@nlsde. buaa. edu. cn.

分布于几乎所有的学科领域;深度 web中的数据大多是结构化、高质量的.

在传统搜索引擎技术的基础上,设计和实现新的算法、技术和系统,更快、更准确的获得这些深度 web资源信息是论文研究的重点.

1 系统框架

图 1是国家科技基础条件平台门户应用系统中所实现的深度 web资源搜索系统总体框架图.通过爬虫系统爬取网页表单[2],以机器学习的方式进行分类判断,判断是否是深度 web资源的入口[3],保存相关元数据信息,形成深度 web资源库;在获得深度 web资源入口的基础上,通过查询关键词分析和管理、页面自动导航,获取深度 web资源结果页面,与此同时结合网页结构化数据抽取,将深度 web页面资源进行结构化处理,获取结构化的深度 web资源信息;在此基础上,结合导航、检索技术,最终实现深度 web资源搜索服务.

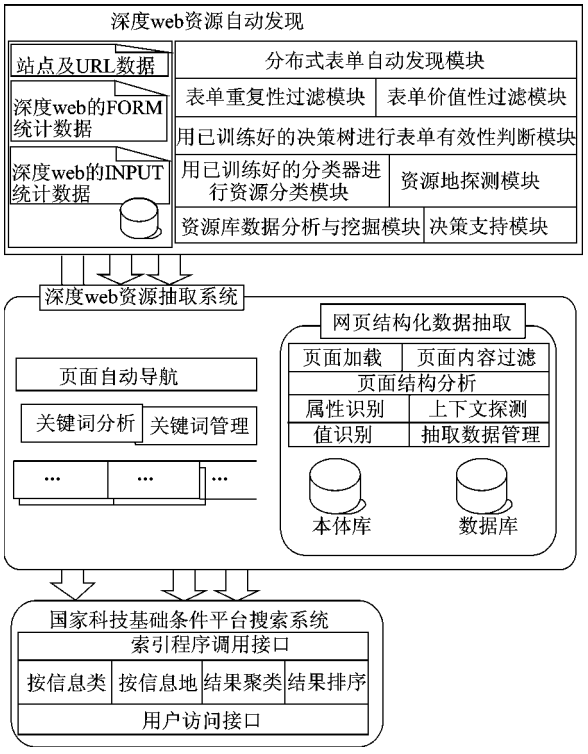

图 1 深度 web资源搜索系统总体框架图

目前国内外在深度 web方面的研究还处于初始阶段,重点还关注在个别点上,没有系统的研究.与该系统最为接近的是美国伊利诺斯州大学的 MetaQuerier项目[4],然而 MetaQuerier项目的研究主要从深度 web资源整合的角度,而且 MetaQuerier项目在深度 web资源搜索方面没有提出一套系统级的解决方案.因此,该深度 web资源搜索系统是目前唯一公开发表的深度 web资源搜索解决方案,并且在国家重大工程专项中进行了实现与应用.

深度 web资源搜索系统可以分为深度 web资源自动发现和基于本体的深度 web资源抽取两大部分组成,下面将介绍这两大部分中的核心关键技术.

2 关键技术

深度 web资源发现的核心是一个网络爬虫,该网络爬虫的目标是尽可能地发现分布在各站点中的 web表单 (候选 web资源入口).深度 web资源发现的网络爬虫宜采用基于站点的爬虫策略.

2.1 基于站点的爬虫策略

基于站点的爬行策略是按照站点之间的链接结构选择爬行目标,爬行过程考虑统一资源定位符 (URL,Uniform Resource Locator)所属站点这一因素,对提取出来的站点之间的关系进行分析.

基于站点爬行的策略具体工作流程为:爬虫在爬行网页的过程中,种子 URL由当前所爬行的页面提供,如果种子 URL属于当前网站内 URL,爬虫将会把 URL提交给当前线程的站内 URL控制器保存处理,如果不是当前站内的 URL,爬虫将会把 URL提交给站点控制器处理.区分种子 URL是否为当前站点的 URL采用对当前站点链接进行分析,提取站点信息中心词的方法,通过匹配站点中心词的策略完成.

2.2 自动域名学习算法

基于站点爬行策略的关键问题之一是如何在大量的网页链接中判断提取出属于当前站点内的网页链接,判断的方法有许多种,比较常用的有通过域名系统 (DNS,Domain Name Sytem)查询链接对应的 IP地址进行判断.基于 IP地址的判断对于大多数网站而言能有效地对网页归属进行判断,但是存在的问题是并不是所有的网站都只拥有一个 IP,随着互联网的发展,网站的访问压力越来越大,多数网站采用了负载均衡的策略,同一个 URL地址在用户进行访问的时候会根据一定的策略将请求转交给不同 IP地址的主机,因此基于 IP地址的访问在这种情况下无法做出准确的归属判断.

系统采用提取中心词的方法对 URL进行站点分析,中心词表示能代表此站点的词语.

系统采用一个自动域名学习算法进行域名的学习,通过对域名的判断确定中心词的位置.域名学习模块算法流程如图 2所示.

通过试验对 1 000个站点域名进行测试,仅

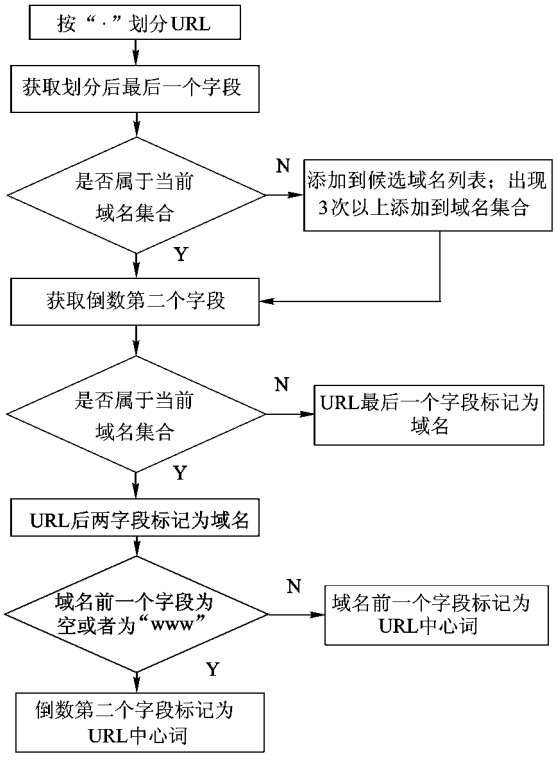

图 2　域名和中心词学习算法流程图

有 2个域名没有分析正确,分析正确率为 99. 8%,并且随着域名的不断学习,分析的正确率会不断增加.

2. 3　实验分析

实验选取了表 1所示的 8个门户型或网站目录型的站点域作为系统爬行的起点,测试基于站点的爬虫算法爬取站点的速度.

表 1　实验所选择的站点

序号	站　点
1	http: //www. edu. cn
2	http: //www. net. cn
3	http: //site. yahoo. com. cn
4	http: //www. baidu. com
5	http: //www. gov. cn
6	http: //www. hao123. com
7	http: //www. vnet. cn
8	http: //www. google. com

系统在前 10 min每分钟取 1次系统数据,初始运行种子域 URL的获取状况如图 3所示.

从图 3可以看出,在系统运行初期,系统获取的种子域 URL基本呈线形递增. 平均 694. 4站点(域) /min.

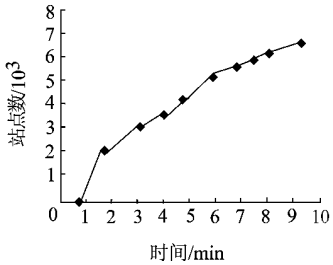

图 3　10 min爬取站点数

3　基于本体的数据抽取关键技术

目前从网页中抽取数据一般是采用包装器的方法,也就是说通过生成包装器,对网页中感兴趣的数据进行抽取.

包装器生成按照自动　　程度可以分为[5]: ①手工构造的包装器,包括 TSIMMIS, W4F和 XWRAP; ②有指导的包装器生成,包括 SRV, RAPIER, STALKER, SoftMealy等; ③半指导的包装器生成,包括 IEPAD, OLERA和 Thresher; ④自动包装器生成,包括 DeLa, RoadRunner, EXALG, DEPTA等.

然而,在包装器抽取网页数据中存在包装器的准确性和维护问题,与此同时基于包装器的数据抽取过程中标签与值的匹配是一个相对复杂的难题.

鉴于基于包装器的网页数据抽取的一些不足,在深度 web资源搜索系统中采用一种基于本体的资源数据抽取技术.

3. 1　数据抽取本体库

数据抽取本体库可以看成是三元组 (O, Φ, Δ). O代表对象集合, $\forall o \in O$

$$o:: = (\mathrm{D}, \mathrm{name}, \mathrm{aliaslist}, \phi, M)$$

其中, D为对象; name为对象名称; aliaslist为对象别名集合; $\phi \in \Phi$,为数据帧; $M \subset \Delta$,为对象的关系集合.

Φ代表数据帧集合. 数据帧是抽取本体的核心,它的本质是定义特定概念的对象集的识别规则,这些规则主要包括正则表达式、宏定义、词典、和值、识别词表达式等方式.

Δ代表关系集合. 关系集合定义了本体库中 2个对象集合之间关系的集合. 关系集定义了概念之间是如何相互联系在一起的. 在本体关系中,很多二元关系集属于拥有 (has)关系,可以用“X has Y”表示,把没有特别指明的关系都默认为这种“has”关系. 描述的主要关系包括关联关系、泛化与具体关系等.

3. 2　文档预处理

对于隐藏于某个查询接口之下的深度资源而言,有价值的数据项往往是以记录集的形式出现在查询结果页面中,因此深度搜索采集到的页面也有按照节点形成簇群分布的特点. 同一簇群的页面也具有一些显著的特征. 这些页面往往包含一些相似度非常强的部分,对于信息提取没有价

值,比如导航、版权信息、头尾等信息、广告.而另外包含了核心数据的部分,则在内容相似度上表现的比较差.

首先对这些页面进行结构分析,将其解析为一颗 Dom 树,再对这些页面的各子树进行结构相似度分析.利用页面间的内容相似度分析,把页面间包含的相似度很强的内容认为是无价值的共同部分,在预处理中加以过滤.同时将相似度差的部分作为数据提取的主题.从而可以从树中提取出来做下一步的结构化数据提取的输入.

匹配 2颗 Dom 树的相似结构,实现文档的预处理的基本思路是使用深度优先的顺序遍历 2颗树,并且从树根到树叶逐个节点进行匹配.

文档预处理算法相关的符号和函数说明: $Root_t$, $Root_\tau$为 2个相似的 Dom 节点; Equal(node n, node m),对于不同节点类型节点含义的解释如表 2所示; Children(node)为 Dom 节点 node的孩子节点数; Child(node i)为 Dom 节点 node的第 i 个孩子节点.

表 2 Equal函数的含义解释

节点类型	函数含义	函数返回值
Attr	节点名和节点值都相同	相同返回 TRUE,否则 FALSE
Text	文本值相同	相同返回 TRUE,否则 FALSE
CDATASection	节点值相同	相同返回 TRUE,否则 FALSE

文档预处理算法 (DataClean)

1) node1 $Root_t$, node2 $Root_\tau$;

2) 如果 Equal(node1, node2) = FALSE,那么返回 False;

3) If(Children(node1) = Children(node2)) > 0
Then
For w: = 1, …, m do
For v: = 1, …, n do
If (DataClean (child (node1, w), child (node2, v))
Removechild(node1, w);
Removechild(node2, v);
If(node1和 node2的所有孩子节点都相等)
Then return True; Else return False

4) If Children(node1) = Children(node2) = 0
Then return True

5) return False

3.3 数据抽取技术

经过上述一系列文档预处理后,文档中保留下来的数据是感兴趣的信息.下一步的任务就是如何从这些主要由超文本标记语言 (HTML, Hyper TextMark-up Language)形式的数据中抽取出结构化数据来.

采用基于本体的数据抽取技术来实现结构化数据抽取工作.该技术的实现主要通过对象属性识别、上下文探测、值识别 3个步骤完成.

1) 对象属性识别.对象属性识别是结构化数据抽取的第一步,其作用是在待处理文档中定位将提取属性的位置.在对象的数据帧中定义了识别词表达式,该表达式由能够表示属性特征的词典组成,对象属性识别就是利用该表达式在文档中定位这些识别词的位置.系统会将正在处理的文档中能够与识别词相对应的词典匹配值标记出来,通过记录这些词的位置,来定位本体对象属性在该文档中的位置.

2) 上下文探测.当对象属性识别结束后,对象的所有属性均被定位,下一步进行上下文探测.对于每个属性,总可在文档中出现属性的位置附近找到赋予该属性的值,而属性上下文即是该属性值出现的范围.上下文探测模块就是在属性出现的位置后确定该属性取值的上下文范围,从而为下一步的值提取提供作准备.

一个 HTML 文档通常包含一些表示停顿或者分隔标记,这些标记将整段的文本分隔为表示了一定语义的较小的语言单位,将这些标记通称为分隔符.本文的上下文选取的依据就是这些文本分隔符号.

一个文档中的分隔符通常包括: HTML 标签、自然语言的语句结束符和自然语言的语句停顿符.这些分隔符都有着分隔文本的作用,但是表示停顿的程度并不一样.根据分隔符表示停顿的程度不同将分隔符分为 2类:停顿符和终止符.

停顿符表示上下文选取的停止标记.即从开始位置到停顿符位置为一段候选的上下文范围.属于停顿符的分隔符包括:逗号、分号、“ < td > ”、“ < li > 等.

终止符表示上下文选取的停止标记.即遇到上下文选取的范围不得超过终止符.属于终止符的分隔符包括:句号、“ < tr > ”、“ < div > ”、“ < p > 等.

上下文探测模块还负责维护上下文的位置信息,它提供了可跟踪路径来为下一步的值识别工作提供位置数据信息.上下文探测模块和值识别组件相结合逐步定位属性值的位置.

3) 值识别.值识别是结构化数据抽取的最后

一步,其作用是在属性的上下文环境中寻找该属性的值信息.那么,该功能是通过本体定义中的值表达式实现的.

每个属性值都由一个规则构成,有的由字符串构成,有的由数字构成,有的包含了特殊符号.而本文就是要利用这些属性值的构成规则,将其形式化为一个正则表达式,然后利用该表达式去属性上下文中寻找匹配该表达式的值.该值就是要找的属性的值信息.根据上文本体库定义的介绍已知,在对象的数据帧中定义了值表达式.值表达式就是由抽取规则组成的,该表达式使用几种可能的正则表达式来匹配目标属性在输入文本中的取值.

属性经过上下文探测之后,将属性的取值所在的范围限定在一定的文本区域内.而值识别组件则在该区域内检索符合值表达式定义的匹配字符串,该字符串就是要抽取的属性的取值.

3.4　数据抽取算法

数据抽取算法相关的符号和函数说明:DocSet:文档集合类型;IEOntology:数据抽取本体;DataCleans(DocSet):针对一个文档集合的数据清洗函数,该函数可以通过简单调用 3.2节所描述的 DataClean函数来实现;ProLoc:对象属性识别函数;ConDetect:上下文探测函数;ValueMap:值抽取函数.

数据抽取算法

```
OntoExtract(DocSet docs, IEOntology onto)
    DataCleans(docs);
    For docs[1], …, docs[n] do
    Begin
        ProList = ExtractProperty(onto);
        For j = 1, …, m do
      Begin
        ProLoc(ProList[j]);
        ConDetect(ProList[j]);
        ValueMap(ProList[j])
      End
    End
```

3.5　实验分析

信息抽取系统常用的评测指标包括准确率 P(precision)和召回率 R(recall).

准确率是指抽取出正确的信息数占所有抽出信息数的比值;召回率是指抽取出正确的信息数占所有正确信息数的比值.其中,P和 R均在[0,1]区间内.准确率为抽取的元数据信息的条数占所有需抽取条数的比例,而召回率为抽取正确的元数据的条数占所抽取条数的比例.数据抽取实验结果如表 3所示.

表 3　数据抽取实验结果表

数据来源	网页数	时间/s	抽取数	P/%	R/%
www.dangdang.com.cn	50	246	50	100	97.0
www.china-pub.com	50	286	49	98	95.0
http://202.112.134.140:8080/opac/	50	192	50	100	99.0
平均值	50	208	50	99.3	97.0

产生误差的原因可能有:某些动态页面生成时由于对方 web服务器工作流量的问题造成页面生成不完全;爬虫爬取过程中可能遇到网络拥塞造成页面下载时丢失部分页面数据;属性识别词表达式的定义不完整,即词典包含的词汇集合不完备;值匹配规则抽取的不够完整,造成值表达式不完备.由前 2种原因造成的误差可以通过改善网络条件改进,后 2种原因造成的误差可以通过持续不断的改进、提炼对应的正则表达式或词典达到改善的目的.

4　结束语

本文提出了一种基于站点的爬虫策略及算法,经过试验分析,该算法针对深度 web资源发现目标进行了优化,算法可行并且高效;提出了一种新的用于数据抽取目标的本体扩展,并提出了基于该数据抽取本体的数据抽取算法,通过实验分析,该算法具有很高的准确性;描述了一个完整的深度 web资源搜索系统.该系统已经在国家重大工程项目中得到应用,并取得良好的应用效果.

参考文献 (References)

[1] Chang K C C, He B, Li C, et al. Structured databases on the web: observations and implications[J]. Sigmod Record, 2004, 33(3): 61 - 70

[2] Chen Peng, Li Tao, Wei Kun, et al. Research on automatic discovery of deep web[J]. Compute Science, 2007, 34(11A): 32 - 35

[3] Chen Peng, Su Liliang, Wei Kun, et al. Towards automatic discovery of deep web based on machine learning[J]. Journal of Computational Information Systems, 2007, 3(3): 1033 - 1042

[4] Chang K C C, He B, Zhang Z. Toward large-scale integration: building a metaquerier over databases on the web[J]. CIDR 2005: 44 - 55

[5] Chang C H, Kayed M, Girgis M R, et al. A survey of web information extraction systems[J]. IEEE Transacitons on Knowledge and Data Engineering, 2006, 18(10): 1411 - 1428

(责任编辑:吴　琼)