WebLight——一个集文档管理的综合搜索引擎

王 亭 赵轶群 (同济大学计算机科学与技术系 200092)
秦 耕 (同济大学管理信息系 200092)

摘要:WebLight 综合搜索引擎提供了统一的查询语法界面,采取多线程机制从指定链接的多个搜索引擎中获取信息。同时,WebLight 提供一个基于目录分类的页面文档管理工具,使得用户更加有效地查询信息。

关键词:搜索引擎 综合搜索引擎 超文本传输协议 公用网关接口 多线程 查准率 查全率 目录分类

一、背景介绍

搜索引擎的发展由于其本身的局限性,限制了信息的更有效的获取。首先,没有一个搜索引擎能够覆盖所有的 WWW 资源,而且大部分搜索引擎的索引平均只能涉及到整个 WWW 资源的 30－50％左右(Search Engine Watch 数据[2])。而国内的几个中文搜索引擎,大部分只对中文页面进行索引,而且,就其中文页面来说,覆盖率也很低,搜索中文页面一般只能在 30 万左右,仍然不能满足网络用户对中文信息的需求。在这样低的信息的覆盖率下,搜索引擎的查全率就不可能达到理想要求。

其次,搜索引擎由于使用了不同的索引技术和信息收集技术,使得各自搜索的信息资源的很大程度上具有相当的差异。如果使用单一搜索引擎,是不可能得到所有的 WWW 的相关资源。人们为了寻找目的资源,往往需要在几个不同引擎间切换,既麻烦,又影响用户的投资。

最后,搜索引擎的发展本身也反映了 WWW 资源的无序性。在搜索关键字查询的句法上,几乎每个搜索引擎都有所不同。同样使用通配符,Altavista 使用“*”,而 Lycos 却用“$”来代替。尽管使用最简单的“AND”、“OR”等布尔搜索,也有一些搜索引擎不支持。用户为了使用这些搜索引擎,将不得不去熟悉每个引擎的不同搜索句法和功能。

因此,本课题立项的动机就是在信息获取的食物链中,开发一个能满足用户同时对性能、通用、简易三种要求的信息获取方式。于是,项目最终定位于一个中文化的综合搜索引擎,并且克服同类系统的不足,尤其是增加了用户急需的分类文档管理功能。

二、相关课题

1. 搜索引擎(Search Engine)

综合搜索引擎是建立在搜索引擎之上的,因此研究一般搜索引擎的机制在本课题中显得异常重要。所谓搜索引擎,就是指能够对自动对 WWW 资源建立索引或进行主题分类,并通过查询语法为用户返回匹配资源的系统。也称为 Spider,Crawler,Indx 等等。

同时,在与搜索引擎发展的同时,也有诸如 Yahoo! 等著名的目录分类服务的门户,这是一种完全由人工对 WWW 资源进行主题分类的门户,也即在上述引擎体系图中“1:Web Robot”部分由手工代替。这样的门户具备精确、可靠的特性,但同时,在及时反映瞬息万变的 WWW 上又不及搜索引擎。

2. 综合搜索引擎(Meta Search Engine)

所谓综合搜索引擎,就是指在统一用户查询界面与信息反馈的形式下,共享多个搜索引擎的资源库为用户提供信息服务的系统。综合搜索引擎与搜索引擎的最大不同在于它没有自己的资源库和 Robots。它以一个代理的角色,接受用户的查询请求,并把查询请求翻译成相对应搜索引擎的查询语法,在分别发出 GCI 请求后,接受多个 HTTP 响应,抽取其中的查询结果以统一形式显示。

斯坦福大学的 Kevin Chen－Chuan Chang 等人 1997 年提出 STARTS 协议,阐述了搜索引擎应遵循的统一接口和统一利用引擎资源的主要问题;协议规范了搜索引擎的界面的格式[2]。Erik Selberg 提出了一个具体的、定位于服务器端的综合搜索引擎 MetaCrawler 模型[3]。

3. 信息事物链和具体获取方式

Web 上获取资源的方式,一般具有三个层次:用浏览器直接从 WWW 获取原始资源、在引擎站点中获取初

步分类资源、利用综合搜索引擎等智能型代理在归整资源中快速获取资源。以上就构成了 WWW 信息资源的食物链以及对应的获取方式，同时也是 WebLight 立项的主要依据。图 1 为 WWW 信息食物链及相应方法：

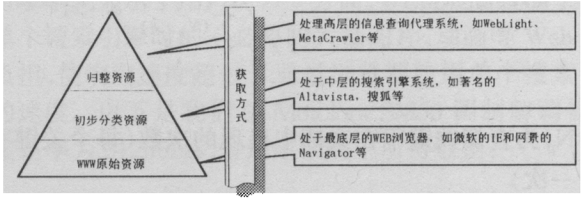


图 1

三、实现思想

1. WebLight 的体系结构图

WebLight 是一个典型的综合搜索引擎，它选择了 AltaVista、中英文 Yahoo、中文搜狐（Sohu）、开网（WebLight）、指南针（Yippee）等著名搜索资源加以综合利用，为用户提供一个统一的查询语法规则和信息反馈界面，同时对下载页面加以集成分类管理。图 2 为其体系结构图：

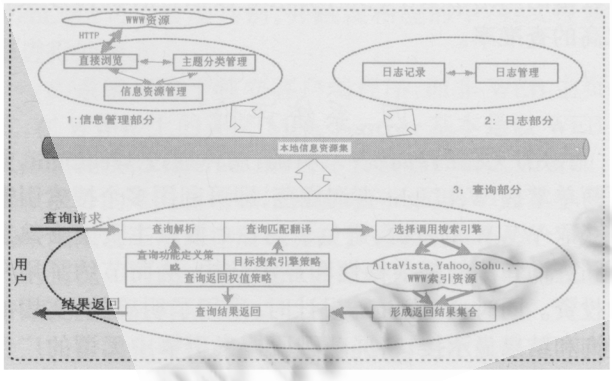


图 2　WebLight 体系结构

2. 实现步骤

从以上的结构图中，WebLight 与一般的搜索引擎的结构相比，有着相似的用户输入与显示接口，只是，在内部实现中，可以根据用户输入特性和预先设置选择预定的搜索引擎资源库，通过 HTTP 协议发送和接受内容。同时，在与查询部分并立的还有文档管理和日志管理部分，其中前者把查询的结果页面或直接浏览保存的页面予以集成管理；日志部分提供了用户查看历史搜索关键字与相应反馈的历史功能。具体来说，一个标准的查询过程，可以有以下步骤：

(1)接受用户查询请求，进行句法分析，并寻找最适合的搜索引擎。

(2)确定搜索引擎后，把查询语句转化为特定的搜索引擎句法。

(3)形成并发送 HTTP 请求，等待接受回应。

(4)处理多个搜索返回结果，以适当的格式和顺序显示。

3. 主要问题

从以上的步骤可以看出，综合搜索引擎需要分析处理所代理的每个搜索引擎的检索策略和语法格式，而且必须能快速的结构返回结果予以显示。但是，由于大部分搜索引擎互相不兼容，相互操作性差，而且用户接口不一致，建立一个强健的综合搜索引擎成为一项复杂的任务。总体来说，有三个主要的问题：

(1)查询语言（QUERY LANGUAGE）的统一问题。这一问题在本文的介绍部分已经阐述。我们的解决办法是：尽可能抽取大多数搜索引擎的共同支持的查询语法格式，形成 WebLight 的查询语法。同时，为了保持各个搜索引擎的不同特性，也支持独自的完全语法。最终，我们提出以下规范格式：

```
@WebLightQuery{
  Version(20): WebLight 1.0
  Stemming(1) : T/F
  StopWords(1): T/F
  CaseSensitive(1): T/F
  QueryFields(20): [Title][+Body][+Author][+
  Date][All]
  WholeWords(1): T/F
  WordsRelevance(3): And/Or
  QueryContents(100):[Sample Text...]
  SearchEngines: [AltaVista][+Yahoo!][+Web-
  Gather][+Yippee][+Sohu]...
  };
```

以上格式采用了 SOIF（Summary Object Interchange Format）[4]规范。其中 WholeWords=T，则对 QueryContents 以整句看待；否则，将 QueryContents 拆分单词，并依据 WordsRelevance 所确定的关系组合，形成 WebLight 语法，然后再解释成由 SearchEngines 所组成的搜索引擎集

合的语法格式,发送到相关的引擎服务器。

(2)HTTP 请求的参数设置(PARAMETER SETTING)问题。为了往各个搜索引擎发送查询请求并且从中分析其中的返回结果,必须精确掌握每个搜索引擎的调用 CGI 格式。如在 AltaVista 中,查询"Tong Ji University"关键词时,该服务器前段将产生以下 HTTP 请求,调用服务器后台的 CGI 程序:

http://www.altavista.com/cgi-bin/query?pg=q&kl=XX&q=Tong%2BJi%2BUniversity&search=Search

WebLight 在接受用户的查询请求后,同样形成以上的 CGI 格式,直接发送到 AltaVista 服务器中,同时等待 HTTP 响应,若响应的格式有多页,还必须接受以下 HTTP 响应:

(Page 2) http://www.altavista.com/cgi-bin/query?pg=q*q=Tong%2bji%2bUniversity&stp=10&c9k

(Page 3) http://www.altavista.com/cgi-bin/query?pg=q&q=Tong%2bji%2bUniversity&stq=20&c9k

(3)多搜索引擎的资源获取(RESOURCE HARVESTING)问题。接受各个搜索引擎的响应后,得到反馈的页面内容后,以下 WebLight 的重要任务将是进行信息检索,确定返回页面资源的地址和主题。这样的主题的任务,要比以下的形成 HTTP 请求要困难得多。因为,不同的搜索引擎的反馈的结果页面的格式相差很大。这部分我们采用了固定查找和智能判断相结合的策略,较好地解决了这一难题。

(4)返回信息的排列(RANK)问题。作为一个综合搜索引擎,信息相关性的排序将是最复杂的问题。这是因为不同搜索引擎在解决这一问题时,采用了相差很大的算法,甚至是一些未知的算法;而综合搜索引擎必须结合这些使用不同排序算法产生的结果,以统一结果形式反馈给用户。

一般的解决办法有两种:

①不加任何排列,直接使用搜索引擎反馈的结果。这样的做法比较简单,但同时由于各个搜索引擎的信息排列的算法的差异,很可能导致信息的无序化。

②在搜索引擎的算法产生的排列的基础上,再进行综合排列,或取所有的搜索引擎的第一条信息形成综合搜索引擎的头几条信息。这种方法是(1)和(3)的折衷,目的是既利用每个搜索引擎的本身排列信息,又兼顾搜索引擎之间的排列的平衡。在追求高速的查询速度为首要目的的综合搜索引擎的开发中,可以采用这种方法。

③得到搜索引擎的返回页面的 URL 地址集合,分别去访问相关的页面内容,然后采用自己的信息相关排列的同意算法,进行最终显示顺序。一个有效的算法如下:

$$R = c_1 N_p + (c_2 - \sum_{i=1}^{N_p-1}\sum_{j=i+1}^{N_p}\min(d(i,j),c_2) \Big/ \sum_{k=1}^{N_p-1}(N_p-k)) \Big/ \frac{c_1}{c_2} + \frac{N_t}{c_3}$$

N_p 为查询关键字在文档中出现的次数(每个关键字只计一次)

N_t 为查询关键字在文档中出现的总次数(每个关键字可以重复)

d(i,j)为第 i 个和第 j 个查询关键字在文档中出现的最短距离(字符数)

c_1 是控制 R 的全局重要性的常量

c_2 是控制有用查询关键字间的最大距离的常量

c_3 是控制关键字频率的重要性的常量

可取 $c_1 = 100, c_2 = 5000, c_3 = 10c_1$

如果查询关键字只有一个,一般可取该关键字在文档中的第一次出现的距离来衡量。上述算法在实际应用中,能较好地量化文档相关性。从而克服每个搜索引擎的比较混乱的排列信息,达到信息的精确定位。但同时在较低的网络带宽下,这种方法是以牺牲查询速度来换取高的查准率。

四、系统评价

1.用户交互界面统一、明确:用户通过 WebLight,只须简单掌握 WebLight 查询规范,即可利用多个搜索引擎来获取 WWW 的资源,避免了到各个搜索主页上去熟悉不同查询句法的复杂特性和显示结构,从而节约了用户的投资。同时,WebLight 设计时,采用了明确的、实用的查询和结果显示接口,屏蔽了原搜索引擎中无谓的广告等冗余信息,方便了用户的浏览。

2.查全率(RECALL)高:高查全率得益于多个搜索引擎的同时使用。在不受限制的情况下,WebLight 的查全率应为:

$R_{wl} = U(P_{av}, P_{yh}, P_{wg}, P_{yp})/p$

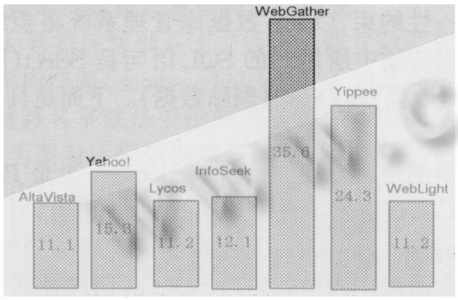
* R=查全率,P=相应的 WWW 页面数,wl=WebLight,av=AltaVista,yh=Yahoo!,wg=WebGather,y=Yippee

从而搜索引擎的区域特性和索引方法来看,

$U(P_{av}, P_{yh}, P_{wg}, P_{yp}) >= Any(P_{av}, P_{yh}, P_{wg}, P_{yp})$

3. 查准率(Precision)高：由于利用了各个搜索引擎的最好的信息排列信息，WebLight 把最相关的信息率先反馈给用户，从而得到了高于一般搜索引擎的查准率。

4. 响应速度(Response Time)快：由于 WebLight 在调用搜索引擎时采取多线程(MultiThread)技术，同时触发每个搜索引擎，并在总控监视器中，监听响应，快的响应引擎率先显示于用户。另一方面，WEbLight 过滤了使用单个搜索引擎时的不必要的大量图片、动画等 Web 资源负担，使响应速度超过了通过浏览器使用单个搜索引擎的速度。以下是我们在 Motolora SM56 调制解调器、连接速率为 31200bps、普通电话线上所得数据：



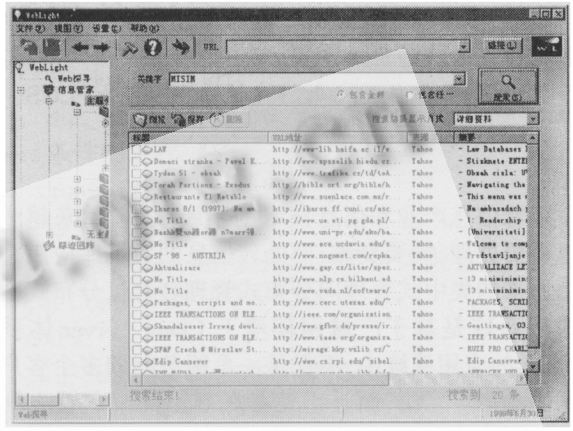
搜索引擎的查询速度的平均值(单位 s)

5. 中/英文的自动处理：根据用户输入的语音种类，WebLight 可以自动判别，并触发相应的中/英文搜索引擎进行处理。

6. 系统定位于独立客户端程序，而非 WEB 主页形式，在 WWW 上的英文 Meta Searcher 基本是以 WEB 主页形式提供给 Internet 用户，尽管使客户群大大增加，但提供服务的本身也增加了带宽争夺，同时 CGI 工作机制的缺陷限制了响应速度的提高。这些尤其不适合我国用户的情况。因此，WebLight 设计的初始就明确定位在客户端的工具软件层次。

7. 分类化文档功能的集成：作为一个成熟可用的信息获取系统，本身不可避免文档的管理和组织。但是现在的 Web 应用程序，大多数只是关心 Web 获取资源的速度，却很少能提供一个切实可行的文档管理工具。比如，在 Internet Explorer，尽管提供了“历史”、“搜藏”等与文档相关的功能，但是过于简单，而且使用不符合用户习惯。因此，我们在 WebLight 中提供了一个类似 Yahoo 的可以无限分类的文档管理工具，在用户的获取相关资源后，能够立即自动存入文档分类中“无主题类”。同时，用户可以用鼠标方便把文档移至事先建立的分类中去。在脱机方式下，点击文档图标，可以在内置的浏览器中浏览上次保存内容。若需要更新内容，用户只须在右键菜单中选择“刷新”即可。分类和文档提供了删除、增加等功能，大大方便了 Web 下载资源的条理化的管理。

以下是 WebLight 总体界面：



分类化的文档管理在画面的左边，检索以及浏览器在右边，用户在检索完毕后可以选中所需页面信息放置到所定类别保存。系统在支持保存页面地址的同时，也保存下载页面信息内容。用户方便时可以离线阅览。

五、不足与展望

以一个实用商品化软件来要求，WebLight 在用户界面等方面仍然有许多不足；尤其在中国实际的应用中，用户实际的操作英文水平较低，而 WWW 上中文信息相对缺乏，这就使得在输入一种语言的关键字时，若进行中英等多种语言查找，将大大提高检索的查全率。同时，中文的全文索引的缺乏，使得查准率上大打折扣。因此，在未来的改进中，建立高效的中文分词系统和中英翻译系统成为关键。

参考文献

[1] Search Engine Watch: http://searchenginewatch.com/

[2] Lambda Luis Gravano, Chen - Chuan K. Chang: STARTS - Stanford Protocol Proposal for Internet Retrieval and Search, ACM SIGMOD 97, 1997

[3] Erik Selberg, Oren Etzioni: Multi - Engine Search and Comparison Using the MetaCrawler, In Proceedings of 4TH International WWW Conference, 1995

[4] SOIF 规范，见 http://harvest.cs.colorado.edu/harvest/user-manual/

(来稿时间：1999 年 7 月)