

基于关键决策方法的站内搜索研究与实现

刘凡平¹, 高艳华¹, 于 炯¹, 张 伟²

(1 新疆大学 软件学院, 新疆 乌鲁木齐 830006; 2 河南省科学院 应用物理研究所, 河南 郑州 450008)

摘 要: 通过对站内搜索的设计分析和实现, 以及对网页排序算法的研究, 充分挖掘网页数据, 提出了新一代站内搜索的关键决策方法. 利用任务分解扩展爬虫, 利用综合倒排文件和快速排序方法进行站内网页决策性排序, 建立新式站内搜索, 并应用于实际. 满足了中小型网站的需求, 充分挖掘了其站内数据, 提升了客户访问量.

关键词: 搜索引擎; 数据挖掘; 网络爬虫; 站内搜索

中图分类号: TP391.9

文献标识码: A

文章编号: 1000-7180(2010)08-0214-04

Research and Realization of Searching System on Website Base on Search Key Decision

LIU Fan-ping¹, GAO Yan-hua¹, YU Jiong¹, ZHANG Wei²

(1 School of Software, Xinjiang University, Wulumuqi 830008, China;

2 Henan Academy of Sciences, Institute of Applied Physics, Zhengzhou 450008, China)

Abstract: Through the analysis on the design and implementation of website searching, the research on web page ranking algorithm and mining the webpage data, a new search key decision is proposed. Using the task decomposition extended spider, comprehensive inverted files and the quick sort method to order the webpages decisively, a new website searching is build and applied for practice to meet the needs of medium sized websites, to fully mine data and to make the clients accessing number increasing.

Key words: search engine; data mining; spider; website searching

1 引言

随着现在网络信息化的迅速普及与发展, 获取网络上面的有用信息成为人们使用的网络的主需求之一, 搜索技术成为达到获取有用信息的主要技术^[1-3]. 设计和开发出一个通用以及具有高效的数据挖掘能力的站内搜索引擎是一个非常具有现实意义的研究^[4].

当前各大网站或门户有站内搜索的绝大多数(尤其是中小型网站)是基于数据库的字段检索实现站内搜索, 这类方法尚有不足之处. 自行开发的全文搜索确实可以在功能上帮助网站解决全文搜索问题, 但也存在时间问题、质量问题、长期维护的问题、经济问题等.

文中设计并实现一种基于关键决策技术的站内搜索引擎. 实现了从大量非结构化、异构的 Web 信息资源中发现有效的、新颖的、潜在可用的及最终可理解的信息. 为中小型网站打造了一个省钱省心省事的站内搜索引擎.

2 设计分析

2.1 系统概述

系统实现分为四个部分: 爬虫部分、分析部分、索引系统、检索系统. 如图 1 所示.

爬虫部分从站点入口开始进行链接的搜集, 存放于链接库中; 分析部分将链接库中链接调动起来获取网页中的数据, 存放于网页内容库中; 索引部分将网页内容库的数据进行分词建立索引, 将索引数

据存放于索引库;用户通过检索系统实现搜索,检索系统先到索引系统中找到对应的网页内容表示,然后从网页内容库中获取数据,返回给用户。

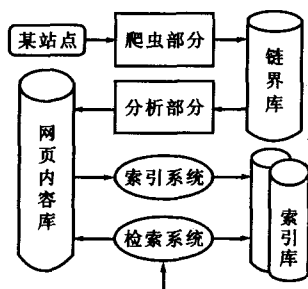


图1 站内搜索系统结构

2.2 爬虫的设计

站内爬虫的主要目的是提取更多的新链接集合,且其具有高度可配置性。提取新的网页链接是利用爬虫程序,利用 Regular Expression 进行 Html 分析获取该入口页面的新链接集合存放于 List 集合中,然后利用多线程递归这个 List 集合,依次将新的 URL 放入 List 集合中,并将每一个 URL 对应的网页下载下来,以方便网页快照使用。如图 2 所示。

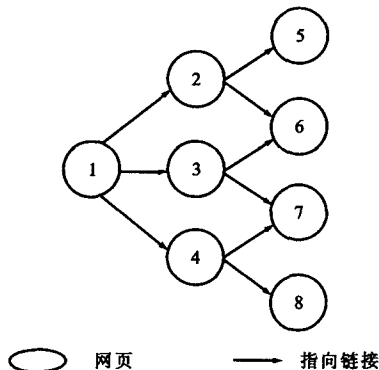


图2 URL 遍历模型

普通的全网搜索^[5]采用宽度优先遍历,而站内搜索根据其特征和需要,采用深度遍历或者宽度遍历均可,因为一个站点的链接是非常有限的。而且站内搜索的一大目的就是充分挖掘站点信息。而深度遍历和宽度遍历均可达到目的。

宽度优先: 1→2→3→4→5→6→7→8

深度优先: 1→2→5→6→3→7→4→8

爬虫将已爬行的数据存放于 Data 文件中,其存放格式如下例所示: < data pageId = "717974241" pageLink = "http://info. xju. edu. cn/index. jsp" pageDesc = "新疆大学, 信息服务" pageTitle = "新疆大学综合信息服务系统" pageSize = "3KB" pageDate = "2010-4-10">新疆大学综合信息服务系

统</data>。

其含义为: pageId 是该信息的标识,是链接 pageLink 的 HashCode. pageId 主要是用于和索引文件的多对多的匹配. pageTitle 是该页面的标题. pageSize 是该页面的大小. pageDate 标识收录日期. Data 节点对应的值是该网页的内容. pageDesc 表示该页面的内容描述即<meta>里面的内容。

在获取内容的时候常常会遇到网页编码的问题,由于各种原因目前的中文编码包括: GB2312-1980、GB18030-2000、GBK、HZ、Big5。一方面获取网页中的 charset 得出网页编码,然后获取网页数据。另一方面采用试探法,以这几类编码分别获取网页数据,利用正则表达式分析是否存在乱码。根据大量实验,如果在前两种方法效果都不理想的情况下,采用 GB2312-1980 进行数据采集,可以取得较好的效果。

爬虫去重也是比较复杂的问题。爬虫记录历史的方式是哈希表,每一条记录是否被抓取的信息存放在哈希表的一个槽位上。如果某网页在过去的某个时刻已经被抓取,则将其对应的槽位的值置为 1;反之置 0^[2]。

2.3 索引的设计

数据结构是一切系统的基础^[1]。建立索引中集中了搜索引擎的两大核心技术难点:索引结构和中文分词。众所周知,索引的结构及存储方法对搜索速度起着致命的影响。经对 Google 和 baidu 的研究以及大量实验的统计,索引结构采用倒排索引。

倒排索引中“倒”的含义是指把关键字和文档以一对多的形式对应起来。倒排序之前的原始文档及分析后的倒排文档如表 1、表 2 所示。

表1 倒排序之前的原始文档

文档编号	关键字
File1	Keyword1, Keyword2, Keyword3, Keyword4, Keyword5...
File2	Keyword2, Keyword3, Keyword4, Keyword5...
File3	Keyword3, Keyword4...
File4	Keyword1, Keyword4...
...	...

在实际物理存储文件中的存储格式如下例所示:< Index KeyWord = "招委会" Count = "6" fileName = "24598_1, 6589_5"/>。表示关键字“招委会”总共出现六次,其中在网页 HashCode 为 24598 的页面中出现了一次,在网页 HashCode 为 6589 的

页面出现了 5 次. 关键字是出现在标题、摘要以及正文中时, 权重是不一样的. 根据大量统计分析, 分别采用权值比例为 10:5:1.

表 2 分析后的倒排文档

关键字	文档编号
Keyword1	File1, File4...
Keyword2	File1, File2...
Keyword3	File1, File2, File3...
Keyword4	File2, File3, File4...
Keyword5	File1, File2...
...	...

实现方法如下:

定义 L 为词库的集合.

定义 l_m 为词库中的第 m 个词汇.

定义 h_m 为网页数据文档 d_i 的哈希值(哈希值是一段数据唯一且极其紧凑的数值表示形式).

定义 K_i 为文档 d_i 的关键字集合.

定义 $t_{i,m}$ 为关键字 l_m 在文档中 d_i 出现的次数.

定义 $l_{j,i}$ 为文档 d_i 中出现的词汇 l_j .

$K_i = \sum l_{j,i}$ (j 为实际值, n 表示 d_i 文件中出现不重复的词汇个数).

定义 P 为词汇 l_m 在所有收录的数据中包含 l_m 的网页集合.

定义 p_j 为词汇在文档 h_m 与出现次数的十万分之一的和.

$$P = \sum p_j \quad (2)$$

$$p_j = h_m + t_{i,j} / 100000 \quad (3)$$

得到的 p_j 是一个浮点数.

对集合 P 进行快速排序. 设要排序的数组是 $p[0] \cdots p[n-1]$, 首先任意选取一个数据(通常选用第一个数据)作为关键数据, 然后将所有比它小的数都放到它前面, 所有比它大的数都放到它后面, 这个过程称为一趟快速排序. 一趟快速排序的算法是:

(1) 设置两个变量 I, J , 排序开始的时候: $I = 0, J = N - 1$;

(2) 以第一个数组元素作为关键数据, 赋值给 key, 即 $\text{key} = p[0]$;

(3) 从 J 开始向前搜索, 即由后开始向前搜索 ($J = J - 1$), 找到第一个小于 key 的值 $p[J]$, 并与 $p[I]$ 交换;

(4) 从 I 开始向后搜索, 即由前开始向后搜索

($I = I + 1$), 找到第一个大于 key 的 $p[I]$, 与 $p[J]$ 交换;

(5) 重复第 3 ~ 5 步, 直到 $I = J$; (3, 4 步是在程序中没找到时候 $J = J - 1, I = I + 1$. 找到并交换的时候 I, J 指针位置不变. 另外当 $I = J$ 这过程一定正好是 $I +$ 或 $J +$ 完成的最后另循环结束)[6].

以上实现了从文档分词后, 建立倒排序的过程, 并且关键字集合得到排序, 以方便实现更高效的查询效果.

另外索引的构造过程中重要部分就是中文分词. 中文分词采用居于词库的逆向最大匹配分词. 其与正向最大匹配分词相反, 是从句子的结尾进行分词. 统计结果表明, 单纯使用正向最大匹配的误差率为 1/169, 单纯使用逆向最大匹配的误差率为 1/245, 显然逆向最大匹配分词法在切分的准确率上比正向最大匹配分词法有较大的提升[3]. 分词效果如下: “淮河以南进入主汛期防汛抗洪形势严峻, 北方地区也即将进入主汛期”.

2.4 检索系统的设计

用户检索之前需启动搜索服务. 搜索服务是使用 WebService 对外提供的一项服务, 外部网页利用 JavaScript 调用这个 WebService 传入的关键字即可, 然后 WebService 将数据返回给页面. 其中在 WebService 返回数据之前需进行网页排序.

当传入参数以后查询的关键词采用二分法查找:

定义 Q 用户输入的查询语句分词后的集合.

定义 q_i 为用户查询语句分词后第 i 个词汇.

$$Q = \sum_{i=0}^{i=n} q_i \quad (n \text{ 为集合个数}) \quad (4)$$

定义 u_i 为 q_i 的哈希值.

定义集合 S 为搜索结果的集合

s_i 为 q_i 在集合 P 中找到的对应网页

流程:

(1) 确定整个查找区间的中间位置

$$\text{mid} = (\text{left} + \text{right}) / 2 \quad (5)$$

(2) 用 q_i 与中间位置的 p_j 进行比较; 若 $|[q_i] - [p_j]| < 1$, 则记录该结果; 若大于, 则在后(右)半个区域继续进行折半查找; 若小于, 则在前(左)半个区域继续进行折半查找[7].

(3) 对确定的缩小区域再按折半公式, 重复上述步骤.

(4) 记录所有的结果集 S .

获取到网页结果集后利用 HashTable 进行整数

部分去重,去重的目的是将多个关键字对应的网页进行整理。

定义 z_i 表示 s_i 的整数部分,同时也可得 $s_i - z_i$ 表示 s_i 的小数部分.利用 HashTable<key, Value> 且 key 为唯一的特性,进行去重,如下伪代码:

```

if(hashtable.Contains( $z_i$ ))
{
    hashtable[ $z_i$ ].value +=  $s_i - z_i$ ;
    //将相同整数部分的值进行小数部分相加,小数部分
    就是出现的次数
}
else
{
    hashtable.add( $z_i$ ,  $s_i - z_i$ );
    //如果没有将其新添加
}

```

定义 A 为 S 去重整理后的新集合。

定义 a_i 是集合 A 中每一个元素的小数部分.将所有的 a_i 根据需要返回的数据条数从高到低进行排序(例如:需要向页面返回 10 条数据,那么只需排前 10 条数据即可).并对应输出其整数部分,整数部分即为 h_m ,即得到的是排序以后的文档 d_i 。

最后根据排序后的文档 d_i 返回数据,包括:标题、数据、网页链接、收录日期、网页大小、网页快照。

3 实验结果及分析

本系统测试实验以新疆大学站点 (<http://www.xju.edu.cn>) 为例,新疆大学站点共有网页量为 2 万余页.如表 3 所示。

表 3 实验结果

输入关键字	返回结果/条	返回时间/s
大学	11827	0.051
大学生活	2580	0.020
校园大学	3501	0.022
校长信箱网址	250	0.005

由以上统计信息可以得出,此站内搜索引擎可

以充分挖掘站点数据,随着关键字的增多,搜索定位越准备,提高了搜索的准确度和搜索速度,提升了网站形象。

4 结束语

文中介绍了一个基于站内搜索的数据挖掘研究与实现,不同于传统的搜索引擎。(1)基于高效的站内搜索,自主实现逆向最大匹配分词,扩展了网页排序方法。(2)利用数据挖掘的原理,实现倒排索引,在各个细节中都优化查找方式。(3)深入细化,准确到位,体贴用户心理,发现用户最关注的内容,寻找用户的潜在需求。

参考文献:

- [1] 卢亮,张博文. 搜索引擎原理、实践与应用[M]. 北京:电子工业出版社,2007.
- [2] 梁斌. 走进搜索引擎[M]. 北京:电子工业出版社,2007.
- [3] 黄魏,高兵,刘异,等. 基于词条组合的中文文本分词方法[J]. 科学技术与工程,2010,10(1):85-89.
- [4] 基于 LUCENE 的站内搜索引擎的设计实现论文[EB/OL]. (2010-05-26) [2010-06-01]. <http://www.souku.com.cn/viewtitle.jsp?url=66541652>.
- [5] 银杏高科. 专业的站内搜索提供商[EB/OL]. (2010-04-20) [2010-06-01]. <http://www.ginkgotek.com/aq2.html>.
- [6] 严蔚敏,吴伟民. 数据结构:C语言版[M]. 北京:清华大学出版社,2005.
- [7] 周娅,张振宇. 数据结构[M]. 重庆:重庆大学出版社,2003.

作者简介:

刘凡平 男,(1988-),研究方向为搜索引擎以及信息处理。
 高艳华 女,(1981-),硕士研究生.研究方向为信息处理与数据挖掘。
 于 炯 男,(1964-),博士,教授.研究方向为计算机分布式系统、网络安全。
 张 伟 男,(1984-),研究实习员.研究方向为电子商务。

(上接第 213 页)

作者简介:

李少雪 男,(1983-),硕士研究生.研究方向为嵌入式系统、网络。

吾守尔·斯拉木 男,(1942-),教授,博士生导师。

郭凯敏 男,(1985-),硕士研究生.研究方向为嵌入式系统、多文种信息处理。