

# 基于 Zipf Estimator 的 Deep Web 最佳查询词选择

王 贤, 苏晓珂, 黄青松

(昆明理工大学 信息与自动化学院, 云南 昆明 650051)

**摘 要:** Deep Web 的查询中, 关键词的选择是一个关键问题。文中针对查询 Deep Web 中的文本数据库, 对查询词的选择作出一些研究。将 Zipf Estimator 应用于根据查询词的频率选择词条的方法中, 提出了用部分文档中的查询词的排序来得出整个文档集中查询词的排序的方法。将 Zipf Estimator 运用于查询词的选择, 减少查询词选择时的运算量, 以较少的查询次数得到较多的查询结果。测试结果证明了 Zipf Estimator 运用于查询词的选择可有效提高查询 Deep Web 中的文本数据库的效率。

**关键词:** Deep Web; Zipf Estimator; 查询词选择

**中图分类号:** TP391

**文献标识码:** A

**文章编号:** 1673-629X(2007)03-0119-02

## Study of Deep Web Query Selection Based on Zipf Estimator

WANG Xian, SU Xiao-ke, HUANG Qing-song

(School of Information Engineering and Automation, Kunming University of  
Science and Technology, Kunming 650051, China)

**Abstract:** In the deep Web inquiry, the key words choice is an important issue. Mainly aims at query words choice when research the text database of deep Web. Zipf Estimator will be applied to the method that base on the frequency of inquiry words choice key word, using listing of member documents inquiry words substitute listing of all documents inquiry words. Zipf Estimator applied to the choice of inquiry words to reduce enquiries word processing volume, received more inquiries results with less inquiries frequency. The test results demonstrated that Zipf Estimator is applied to inquiry words choice can effectively raise speed of inquiry text database of deep Web.

**Key words:** deep Web; Zipf estimator; query selection

## 0 概 述

近年来, 随着在线数据库的流行, 网络正迅速地深化, 研究表明大量的数据隐藏在 Deep Web<sup>[1]</sup>中, 这些数据无法直接通过静态的 URL 连接来获得, 只有通过数据库查询接口动态地提交查询来获得, 即在查询接口中输入关键词来获得站点中的网页。Deep Web 上有着准确率极高的无限的资源<sup>[1]</sup>, 如果用户要查询 Deep Web 上的内容, 就要找到 Deep Web 的站点并且在查询表格中输入关键字, 才能找到想要的内容。在这样的前提下, 即使用户知道一些 Deep Web 的站点并且进行查找, 也要花费很多的时间和精力来访问所有的相关站点并查看结果。有研究者提出构造一个针对 Deep Web 的爬虫<sup>[2]</sup>来自动搜索 Deep Web 上的信息, 将输入关键字、搜索网页、提交结果这样一个过程让 Deep Web 爬虫自动完成。

要构建一个 Deep Web 爬虫算法, 其中关键的一个问题是选择一个最佳的查询关键词。因为, Deep Web 上有海量的数据<sup>[3]</sup>, 如果能找到最佳查询关键词, 就可以用较小的开销从 Web 数据库上获得较多的网页, 提高爬虫算法的效率。如果不能找到最佳查询关键词, 那么对 Deep Web 的搜索将会是一个较为费时并且结果有偏差的过程。

文中采用根据频率选择最佳的查询关键词的方法, 并且用 Zipf Estimator<sup>[4]</sup>对其作了一些改进。用一部分文档集中关键词所占的频率来估计整个文档集中该关键词的频率。根据频率得到关键词的排序, 根据列表中词条的顺序选择查询词对 Deep Web 进行搜索。

## 1 Zipf Estimator

### 1.1 查询关键词的选择的基本思想

设  $S$  是所有文档的集合。将查询列表  $T$  中的查询词  $Q_i$  填入多个 Web 站点的查询接口中, 可以获得发布每个查询词  $Q_i$  所获得的文档数用  $n_i$  来表示, 那么  $n_i$

收稿日期: 2006-05-19

作者简介: 王 贤 (1976-), 女, 山东人, 硕士研究生, 研究方向为智能信息系统; 黄青松, 教授, 研究方向为智能信息系统。



是  $S$  的一个子集。每个子集都有一个权重来代表发布这个查询所花费的开销(在文中的最佳查询词的选择中假设所有词条的开销是一个常数)。选择最优查询词的问题就转换成为选择包含最多文档的子集  $n_i$ , 与图论的集合覆盖相似<sup>[5]</sup>。如果一个查询词在文档中出现的频率最高, 那么用这个词来填入查询接口来查询 Web 站点将得到最多的网页。

选择某个 Web 站点, 该站点共有文档数为  $N$ , 假设至少已知三个查询词在  $N$  中的频率, 从查询词列表  $T$  中选择查询词  $Q_i$  并发到 Web 站点上, 得到查询词的分布频率和返回的文档数, 通过计算得到总的文档数  $M$  和每个查询词在  $M$  中的频率, 将这些频率排序得到排序  $R(Q_i)$ 。根据  $R(Q_i)$ , 通过公式计算出  $T$  中每个查询词  $Q_i$  在  $N$  中的频率, 并将其按降序排列, 得到一个新的查询词列表  $L$ 。

### 1.2 估计查询词的频率

通过查询词在文档子集中的频率来估计一个查询词在所有文档中出现的频率。在文本文档中关键词出现的频率服从幂次分布<sup>[6]</sup>, 即将所有的关键词依据出现的频率排序, 则一篇文档的关键词的分布:

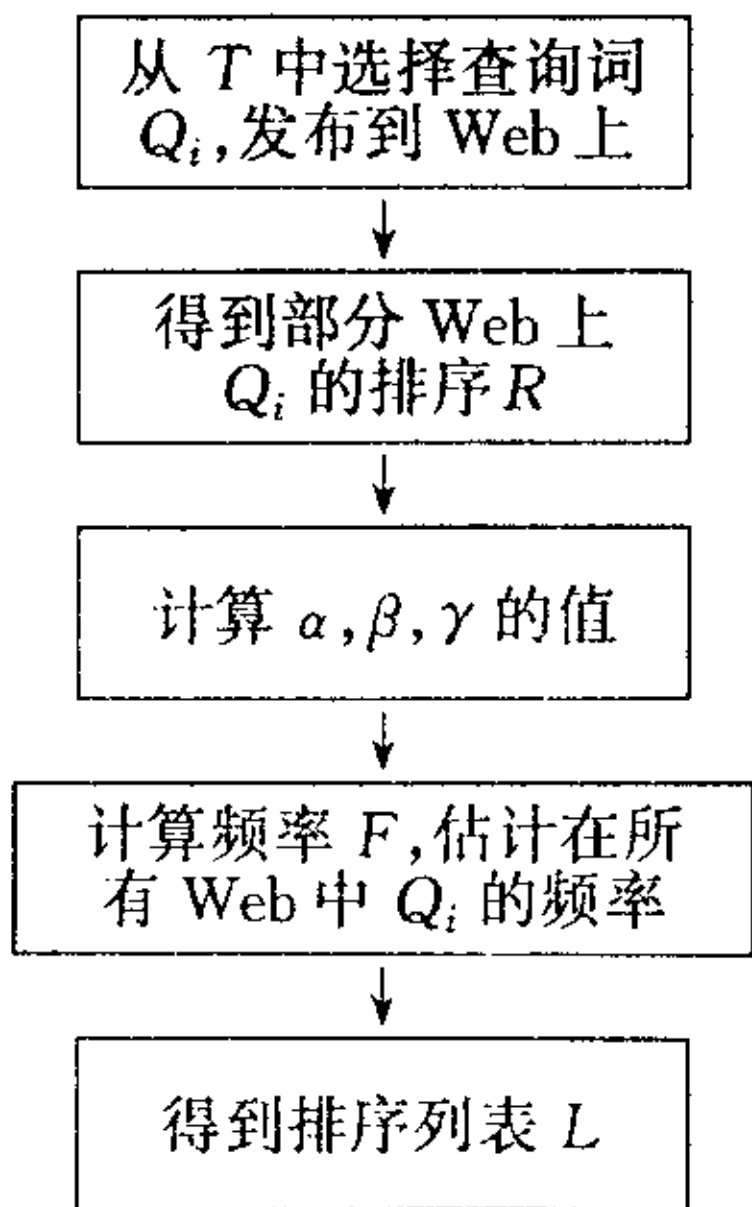
$$f = \alpha(r + \beta)^{-\gamma}$$

其中,  $f$  是关键词在整个文档  $N$  中的频率,  $r$  是词条的排序序号, 根据词条在已下载的文档  $M$  中的频率来排序, 频率最高的词条序号为 1, 次高的为 2, 依次类推。 $\alpha, \beta, \gamma$  是与文档相关的常量。

## 2 将 Zipf Estimator 用于最佳查询词选择的算法

### 2.1 流程图

$T$  为已知的查询词列表, 流程图如下所示:



### 2.2 算 法

输入查询词列表  $T$ , 已知  $T$  中至少三个查询词在  $N$  篇文档中的频率。

输出降序排列的列表  $L$  的算法如下所示:

```

for each  $Q_i$  in  $T$  do ( $Q_i$  是  $T$  中的查询词)
    Select  $Q_i$  with maximum  $n_i$  to send to the Web
    get  $n_i$ 
Endfor
get  $R$ 
get  $\alpha, \beta, \gamma$ 
for each  $Q_i$  in  $T$  do
    get  $f(Q_i)$ 
Endfor
get  $L$ 
    
```

## 3 算法试验

### 3.1 实验设置

在实验中, 列表  $T$  中的查询词采用的是在计算机软件这个类别中的部分查询词,  $T = \{\text{办公, 商务, 计算机, 财务, 工具, 游戏, 教学}\}$ , 仅对三个网站进行试验。第一次将  $T$  中的每个查询词发布三个网站上, 得到  $f(q_i)$  和排序列表  $L_1$ 。第二次  $T$  中的三个查询词的频率为在整个文档中的频率  $f(q_i)$ , 其他查询词的频率用公式来计算, 得到排序列表  $L_2$ , 看  $L_2$  中查询词的顺序是否与列表  $L_1$  的相同。

### 3.2 实验结果

在三个网站中来发布  $T$  中的查询词:

- \* <http://www.joyo.com/>
- \* <http://www.pcsoft.com.cn/>
- \* <http://www.hhsoft.com.cn>

将  $T$  中的每个查询词发布在三个网站上, 得到  $f(q_i)$  和排序序列  $L_1$ , 如表 1 所示。

表 1 在所有文档中查询词的排序结果

查询词	文档数 $n_i$	文档数 $n_i$	文档数 $n_i$	$f(q_i)$	$L_1$
办公	19	170	11	0.073	6
商务	22	111	6	0.051	7
计算机	47	458	14	0.19	3
工具	115	617	16	0.274	1
财务	8	305	16	0.12	4
教学	177	76	8	0.095	5
游戏	248	230	48	0.193	2

$L_1 = \{\text{工具, 游戏, 计算机, 财务, 教学, 办公, 商务}\}$

假设已知在  $N$  篇文档中频率的查询词为: 办公、商务、计算机。要计算其他的四个查询词在  $N$  篇文档中的频率。通过表 2 得到  $R = \{\text{游戏 1, 工具 2, 教学 3, 计算机 4, 办公 5, 商务 6, 财务 7}\}$ , 运用公式:  $f = \alpha(r + \beta)^{-\gamma}$  得到  $\alpha, \beta, \gamma$  的值分别为: 6.1, 1.19, 2.67, 得到四个查询词在  $N$  篇文档中的频率。如表 2 所示, 经过计

(下转第 124 页)



①它用一种高度压缩的结构存储数据库中所有有意义的信息,仅需扫描两次数据库;

②它将长的频集分割成多个长度为 1 的频繁项,一步一步用模式生长的方法产生较长的频集,避免了大量候选项集的产生和测试过程,直接产生频繁模式。

HCS-Mine 算法同样无需产生候选项集,且仅扫描一次数据库。和 FP-growth 算法相比,HCS-Mine 算法采用链表结构:一方面,当数据库更新时,算法更易于扩展;另外,算法不需其他辅助数据结构,尤其对项数相对较小,数据库又很大的情况,算法效率占优。

### 3 结束语

文中分析比较了几种典型的关联规则挖掘算法,此外,发现最大频繁项集也是关联规则挖掘的一个重要算法思想<sup>[11]</sup>:最大频繁项集集合中隐含了所有频繁项集,因此可把发现频繁项集的问题转化为发现最大频繁项集的问题,该方面的研究工作尚不太充分。与此同时,在对关联规则挖掘问题进行大量研究工作的基础之上,研究人员还提出了一些相关的变体算法,如泛化的关联规则、周期关联规则等。

#### 参考文献:

[1] Agrawal R, Srikant S. Fast Algorithms for Mining Association

Rules[C]//VLDB'94. Santiago, Chile: [s. n.], 1994: 487 - 499.

[2] Park J S, Chen M S, Yu P S. An Effective Hash - Based Algorithm for Mining Association Rules[C]//SIGMOD'95. San Jose, CA: [s. n.], 1995: 175 - 186.

[3] 杜孝平, 马秀莉, 唐世渭, 等. 快速关联规则挖掘算法[J]. 计算机工程与应用, 2002(11): 1 - 4.

[4] 郭景峰, 路 燕. 一种数据挖掘关联规则的高效算法[J]. 燕山大学学报, 2001, 25(3): 213 - 216.

[5] 顾泽元, 吕宗宝, 刘兴丽. 频繁项目集发现算法 Apriori 的研究[J]. 黑龙江科技学院学报, 2005, 15(5): 319 - 322.

[6] DE-CHANG P I. STBAR: A More Efficient Algorithm for Association Rules Mining[C]//MLC'05. Guangzhou: [s. n.], 2005: 18 - 21.

[7] 牛小飞, 石 兵. 基于向量和矩阵的挖掘关联规则的高效算法[J]. 计算机工程与应用, 2004(12): 170 - 173.

[8] Han J, Pei J, Yin Y. Mining frequent patterns without candidate generation[C]//SIGMOD'00. Dallas, TX: [s. n.], 2000: 1 - 12.

[9] 叶飞跃, 王建东, 陈慧萍, 等. 基于哈希链结构的频繁模式挖掘[J]. 计算机工程与应用, 2004(11): 174 - 176.

[10] 邵峰晶, 于忠清. 数据挖掘原理与算法[M]. 北京: 中国水利水电出版社, 2003: 158 - 162.

[11] 李 超, 余昭平. 基于最大模式的关联规则挖掘研究[J]. 微计算机信息, 2006, 22(2-3): 164 - 165.

(上接第 120 页)

算,  $f(q_i)$  的值发生了变化, 可是频率的相对顺序没有发生改变, 频率较高的查询词排在前面, 频率低的排在列表后面。

表 2 运用文中的方法排序后的结果

查询词	文档数 $n_i$	文档数 $n_i$	$P(q_i)$	$f(q_i)$	$L_2$
办公	19	11	0.039	0.073	6
商务	22	6	0.037	0.051	7
计算机	47	14	0.08	0.19	3
工具	115	92	0.273	0.273	2
财务	8	16	0.03	0.074	5
教学	177	8	0.245	0.132	4
游戏	248	48	0.39	0.746	1

$L_2 = \{\text{游戏, 工具, 计算机, 教学, 财务, 办公, 商务}\}$

比较两个表, 可以看出经公式计算后, 列表  $L_1$  与  $L_2$  中查询词的排序相对顺序是吻合的。即可以用部分文档中的查询词得出整个文档中查询词频率的规律。

### 4 结束语

查询词的选择是 Deep Web 搜索的一个关键问题, 能有效地找到适合的查询词来填入查询接口, 可以提高 Deep Web 的效率。文中将 Zipf Estimator 应用于根

据查询词的频率选择词条的方法中, 提出了用部分文档中的查询词的排序来得出整个文档中查询词的排序的方法, 对提高 Deep Web 的搜索效率有积极作用。今后的工作重点是在更大范围中进一步研究这种方法, 并不断地完善。

#### 参考文献:

[1] Chang K Chen - Chuan, He Bin, Li Chengkai, et al. Structured database on the web: Observations and Implications[J]. SIGMOD Record, 2004, 33(3): 61 - 70.

[2] Raghavan S, Garcia - Molina H. Crawling the hidden web [C]//In VLDB. Roma, Italy: [s. n.], 2001.

[3] Bergman M K. The deep web: Surfacing hidden value[EB/OL]. 2001 - 07. <http://www.press.umich.edu/jep/07-01/bergman.html>.

[4] Ipeirotis P G, Gravano L. Distributed search over the hidden web: Hierarchical database sampling and selection [C]//In VLDB. Hong Kong, China: [s. n.], 2002.

[5] Cormen T H, Leiserson C E, Rivest R L. Introduction to Algorithms[M]. 2nd Edition. [s. l]: MIT Press/McGraw Hill, 2001.

[6] Zipf G K. Human Behavior and the Principle of Least - Effort [M]. Cambridge, MA: Addison - Wesley, 1949.