Google 搜索引擎技术实现探究

化柏林

(中国科学技术信息研究所 北京 100038)

【摘要】 从技术的角度剖析了 Google 搜索引擎的体系结构与工作过程, 详细介绍了基于 Robot 的网页搜索、标引入库和检索引擎三大模块, 统计了 Google 的技术数据, 并分析了 Google 的技术实现特点, 解释了 Google 检索的种种现象。

【关键词】 Google 搜索引擎 技术实现 【分类号】 G354

Anatomy of Google Search Engine Viewed on Technical Implementation

Hua Bolin

(*Institute of Scientific and Technical Information of China, Beijing 100038, China*)

【**Abstract**】 This paper anatomizes architecture and procedure of Google viewed on Technical Implementation. It introduces three functional modules, which is Web crawler, index and create database, search engine. Then do a statistic of technical data about Google, and analyzes technical feature, explaines a variety of phenomena when using Google to retrieval.

【**Keywords**】 Google Search engines Technical implementation

1 Google 技术总况与体系结构

Google 拥有 10 亿个网址, 30 亿个网页, 3.9 亿张图像, Google 支持 66 种语言接口, 16 种文件格式, 面对如此海量的数据和如此异构的信息, Google 是如何实现半秒内搜索的呢? Google 拥有 1600 台服务器, 大部分代码用 C 或 C++ 实现, 有很好的执行效率, 运行在 Solaris 或 Linux 上。Google 用了 64 个桶(Barrels), 有 293M 词典(Lexicon)、43G 的顺排档文件、41G 的倒排档文件, 构造了一个 5.18 亿个超链接的网络关联图。

Google 搜索引擎有两个特征来提高查准率: 利用网页间的链接关系来计算每一个网页的等级; 利用链接关系来改善检索结果。除此之外, Google 还对所有的点击都有定位信息, 广泛利用搜索的亲近度。Google 记录详细的可视化表达诸如词的字体大小, 大或粗体的词的权重就高。整个页面的 HTML 源文件在知识库中是可用的。

Google 搜索引擎从功能上同样分为三大部分: 网页爬行、标引入库和用户查询。网页爬行主要负责网页的抓取, 由 URL 服务器、爬行器、存储器、分析器和 URL 解析器组成, 爬行器是该部分的核心; 标引入库主要负责对网页内容进行分析, 对文档进行标引并存储到数据库里, 由标引器和分类器组成, 该模块涉及许多文件和数据, 有关于桶的操作是该部分的核心; 用户查询主要负责分析用户输入的检索表达式, 匹配相关文档, 把检索结果返回给用户, 由查询器和网页级别评定器组成, 其中网页等级的计算是该部分的核心。其总体系统结构如图 1 所示。

收稿日期: 2004-03-04

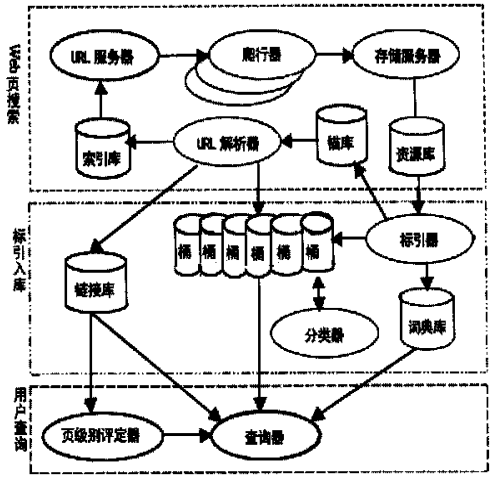

图 1 Google 体系结构图

2 基于 Robot 的搜索过程

Robot 使用多线程并发搜索技术, 主要完成文档访问代理、路径选择引擎和访问控制引擎。基于 Robot 的 Web 页搜索模块主要由 URL 服务器、爬行器、存储器、URL 解析器四大功能部件和资源库、锚库、链接库三大数据资源构成, 另外还要借助标引器的一个辅助功能。具体过程是, 有个 URL 服务器发送要去抓取的 URL, 爬行器根据 URL 抓取 Web 页并送给存储器, 存储器压缩 Web 页并存入数据资源库, 然后由标引器分析每个 Web 页的所有链接并把相关的重要信息存储在 Anchors 文件中。URL 解析器读 Anchors 文件并解析 URL, 然后依次转成 docID。再把 Anchor 文本变成顺排索引, 送入索引库。具体过程如图 2 所示。

2.1 URL 服务器(URL Server)

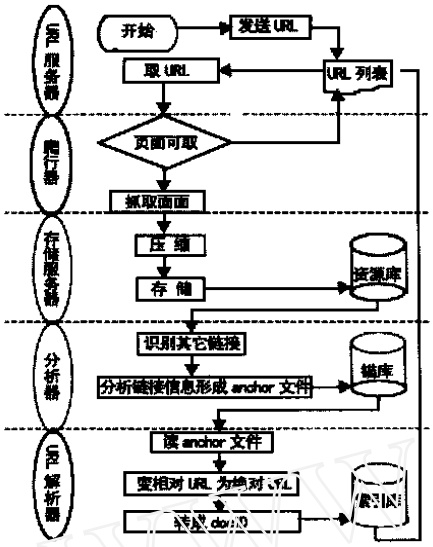

图 2 基于 Robot 的 Web 页搜索过程图

URL 服务器是整个 Web 页搜索模块的开始, 主要用来管理和维护 URL 列表。首先由它发送一个新的 URL 给爬行器, 让爬行器去搜索。如果爬行器遇到了不可下载的网页, 就会给 URL 一个返回信息, 然后取下一个 URL。URL 服务器会从文档索引库里不断地取新的 URL 以供爬行器使用。

2.2 爬行器(Crawler)

爬行器是整个搜索模块中最关键的一部分, 它由好几个分布的爬行器组成, 并协同工作。爬行器遇到 HTML 页的头有如下标记就不再抓取此页, < head > < meta name = "robots" content = "noindex, nofollow" > < /head > , 返回一个空值, 继续向其他方向爬行, 这就有效防止爬行器标引此页及本页的相关链接页。如果网页已经标引过, 就从将要爬行的网页队列中移除。Web 页文本的繁殖思想是由 3W 蠕虫来实现的, 当它搜索非文本信息时, 尽可能少的下载文档, 以扩展搜索度, 因为非文本信息如图片等没有链接会造成爬行的中断。这样, Google 就可以通过各种策略来解决排序沉没(Rank sink)和排序漏出(Rank loak)等问题。

2.3 存储器(Store server)

存储器把爬行器抓来的 Web 页进行压缩并存储到数据资源库中。数据资源库包含每一个 Web 页的全部 HTML, 所有的网页用 zlib 进行压缩, Google 更注重 zlib 的压缩速度而不是压缩率。bzip 对数据库的压缩率接近 4: 1, 而 zlib 为 3: 1。这样, Google 就能把 147.8G 的 HTML 文档压缩成 53.5G 的数据存储在库中。在数据资源库中, 对文档进行归类, 加上 docID 前缀、文档的长度和 URL。文档索引记录着每一个文档的信息, 它是一个有固定长度, 经 docID 排序的 ISAM 索引(Index Sequential Access Mode)。存在每个条目中的信息包括当前文档状态, 数据库中的指针, 文档校验和以及其它统计信息。如果文档被爬行过, 它也包含到可变长文件的指针, 这个称为 Docinfo 的文件包含文档的 URL 和标题。否则, 指针指向仅包含 URL 的 URL 列表。

2.4 分析器(Parser)

分析器可以看成是标引器的一部分, 也可以说是标引器的一个辅助功能部分。它分析每个 Web 页的所有链接并把相关的重要信息存储在 Anchors 文件中, 构成一个锚库。每当从 Web 页分析出一个新的 URL 时, 为每个 Web 页分配一个称为 docID 的关联 ID。这个文件包含足够的信息来决定每个链接的何去何从。锚经常提供比 Web 页本身更精确的页描述。锚可以存在文档, 这些文档不能被基于文本的搜索引擎所标引, 如图片、程序、数据库。这就为返回那些不能精确爬行的 Web 页提供了可能。

2.5 URL 解析器(URL Resolver)

URL 解析器读 Anchors 文件并把相对 URL 转成绝对 URL, 然后依次转成 docID。把 Anchor 文本变成顺排索引, 存到文档索引库里, 并用 Anchor 所指向的 docID 进行关联。把 URL 转换成 docID 的文件, 是由 URL 校验和及相应的 docID 两列组成的一个列表, 并以校验和排序。为了找到一个特定 URL 的 docID, 首先计算 URL 的校验和, 在校验和文件中进行二元查找, 以找到相应的 docID。执行一次批处理, 通过合并文件把 URL 转成 docID。使用这种批处理模式很关键, 要不然就得为每一个链接都作一次查找, 假设一个磁盘上有 322, 000, 000 个链接记录, 那么这样一个过程需要 2 个多月的时间。它还产生成对 docID 的链接数据库, 以用于计算所有文档的 PageRanks。

3 标引入库

标引入库模块由分类器和标引器组成。标引入库模块处理大量的文件和数据, 用来构建庞大的数据库, 主要涉及数据资源库、词典库、链接库、桶等。桶的结构与内容非常复杂, 有关桶的操作是本模块的核心。

3.1 分类器(Sorter)

分类器从桶中取出数据, 按 docID 进行一级分类, 然后按照 wordID 进行二级分类并产生倒排档索引。分类器产生 wordID 的列表并把其偏移量写到倒排档索引中。一个称为 DumpLexicon 的程序把这个列表和由标引器产生的词典揉和在一起并为检索器产生一个新的词典。

3.2 标引器(Indexer)

标引器有许多函数, 它读数据库, 解压缩文档然后进行分析。每个文档都被转成一套单词出现频率, 称之为采样数。采样数记录单词及在文档中出现的位置, 字体的大小以及大写信息。标引器把这些采样数分配到一套"桶"中, 创建一个部分分类的顺排索引。对于中文, Google 主要采用了二元切分法, 也就是为什么我们输入长于两个汉字的中文, 如果不加双引号, Google 会自动给以切分的原因。

3.3 桶(Barrels)

Google 共有 64 个桶(Barrels), 每个桶都存着 wordID 的归类, 包括顺排档与倒排档。如果一个文档包含落在某个桶里的词, docID 和 wordID 的列表以及相应的命中列表就被记录到桶里。Google 存储每一个 wordID 时, 存储的是与所在桶的

最小 wordID 的相对差异, 而不是存储实际的 wordID。这样, 在未排序的桶中用 24 位存储 wordID, 留下 8 位用来存储命中列表的长度。

倒排档索引就像顺排档一样由系列的桶组成, 唯一的不同是被分类器处理过。有个重要的问题是 docID 应当在 Doclist 中如何排序。一个简单的解决办法是用 docID 进行分类, 这允许多词查询而带来的不同 Doclist 的合并。另外一个办法是按词在每个文档中出现的频率等级进行分类存储, 尽管这使得处理单个词的查询变得繁琐, 但为多词查询提供了可能。Google 在这两个方案中选择了折衷, 使用两套倒排的桶, 一套为包括标题和 Anchor hits 的命中列表, 我们称之为短桶, 另一套为所有的命中列表, 我们称之为长桶。

在顺排档索引和倒排档索引中, 命中列表占据了大量的空间。命中列表是指词在一篇文档中的出现频率, 包括位置、字体和大写信息。Google 为编码位置、字体和大写考虑了多种编码方案——简单编码、优化压缩编码和哈夫曼编码。由于优化压缩编码对空间的要求比简单编码低, 操作过程比哈夫曼编码简单, 因此, Google 最终选择了优化压缩编码。

Google 用两个字节的压缩编码来记录每次命中, 命中类型有两种: 特殊命中与普通命中。特殊命中包括 URL 的命中率、标题、链接文本和关键标记。普通命中含有所有的信息, 包括大写位、字体大小和 12 位标记词在文档中出现的位置信息等。字体大小用 3 位来表达文档的其它内容的相对值(仅有 7 个值可用, 因为 111 是特殊命中的标记)。特殊标记由大写位, 字体设成 7 来表明是一个特殊命中, 有 4 位表示类型, 用 8 位标记位置。为了节省空间, 命中列表的长度由顺排档中的 wordID 和倒排档中的 docID 决定。分别需要 8 位与 5 位。如果长度更长的话, 在相应的位里就存着一个溢出编码, 接下来的两个字节存储命中列表的实际长度。

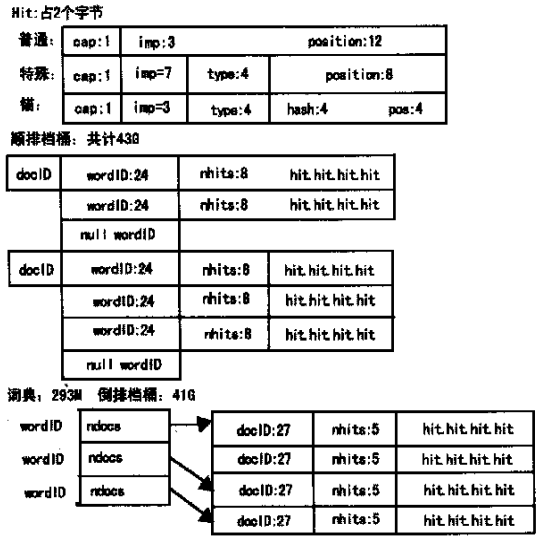

图 3 顺排档索引、倒排档索引与词典

3.4 词典(Lexicon)

词典有几种不同的形式。对早期系统的一个重要改变是根据合理的代价分配内存。该词典包含 1400 万个词条(有些生僻词没加), 由两部分实现, 词表和指针的哈希表。词表还有一些辅助信息用以实现其它功能。对于每个有效的 wordID, 词典包含指向 wordID 所在桶的指针。它指向 docID 和相应的命中列表的 Doclist。Doclist 表明该词在所有文档中出现的所有情况。

4 检索过程与网页级别

Google 用户查询模块主要由网页级别评定器和查询器组成。

4.1 查询器(Searcher)

查询器运行在 Web 服务器上, 并用 DumpLexicon 产生的词典、倒排档索引和 PageRanks 一起来响应查询。首先接受用户输入的检索表达式, 进行分析得出各项检索要求, 提取检索词并转成 wordID, 接着到短桶文档列表里进行查词, 遍历文档列表直到匹配所有的词条, 找到一篇就计算网页等级, 短桶查完了, 如果没有足够的匹配记录, 就去查长桶。查完了所有的文档列表, 就对检索结果进行排序并返回前 K 项。

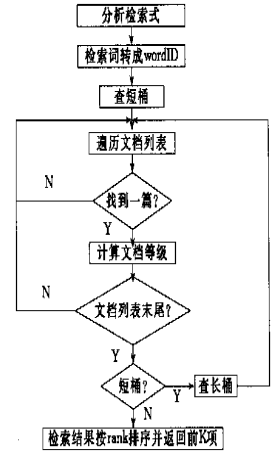

图 4 Google 检索流程图

4.2 网页级别评定器(PageRanker)

网页级别评定器借用了图书文献里的参考文献与引用文献的评价思想, 利用链接网页的数量及重要性进行等级评定, 而链接网页的重要性由它的链接网页的数量及重要性决定, 因此是一种迭代计算。评级函数有许多参数, 如类型权重和相关类型权重等。

如果有许多网页指向网页 P, 那么 P 就是一个好的网页。IR'(P), 从不同域的链接要比同一个域内的链接重要。自然链接可以表明相关重要性。当从网页 A 链接到网页 B 时, Google 就认为"网页 A 投了网页 B 一票"。Google 根据网页的得票数评定其重要性。除了考虑网页得票数(即链接)的纯数量之外, Google 还要分析投票的网页。"重要"的网页所投出的票就会有更高的权重, 也有助于提高其它网页的"重要性"。被引率高就说明该页值得看, PageRank 通过 Web 链接架构来处理靠递归繁殖提高权重的情况。

假定网页 A 有指向 A 的网页 T1..Tn(也叫引用网页)。参数 d 是一个设定于 0、1 之间的递减因子, 通常定为 0.85。C

(A)为从网页A 链出去的链接数,因此,网页A 的 PageRank 就可以求出:

PR(A) = (1- d) + d (PR(T1)/C(T1) + ... + PR(Tn)/C(Tn))

PageRank 或者 PR(A)用迭代算法来计算,相当于一个规范化的Web 链接矩阵的特征向量。有 26,000,000 个网页在一台中型工作站上用几个小时就能算完。PageRanks 形成了Web 页的概率分布,所有Web 页的 PageRanks 和为 1。

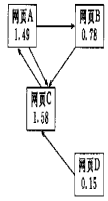

图 5 网页之间的链接关系与 PageRank 举例

重要的、高质量的网页会获得较高的网页级别。Google 在排列其搜索结果时,总会考虑每个网页的级别。当然,如果不能满足用户的查询要求,网页级别再高对用户来说也毫无意义。因此,Google 将网页级别与完善的文本匹配技术结合在一起,为用户找到最重要、最有用的网页。Google 所关注的远不只是关键词在网页上出现的次数,它还对该网页的内容以及该网页所链接的内容进行全面检查,从而确定该网页是否满足用户的查询要求。Google 在检索引擎里用一个用户反馈机,对信任用户可有选择地评估所有的返回结果,这种反馈被存到数据库里,当修改评级函数时,就能发现对以前评过级的检索所产生的影响。

5 功能检索的实现

对于 Google 的检索实现,笔者对检索做了大量的测试,而对技术仅作了有限的部分测试,其理解如下:

(1)逻辑表达式 Google 支持的逻辑运算有与、或、非,其形式分别为" "、"or"、"- ",对应 Google 高级检索里的"包含全部字词"、"包含任何一个字词"、"不包含以下字词"。Google 不支持截词检索,因为截词检索会极大地降低计算机的检索速度。当然 Google 对词的切分技术也并不理想,对西文比较有效,对亚洲语言作得不好。

(2)指定文件类型 Google 可以指定查找的文件格式,如 pdf、doc、ppt 等格式文件。如果某个搜索结果是 PDF 文件而不是网页,它的标题前面会出现以蓝色字体标明的[PDF]。用户只想查一般网页,而不要 PDF 文件,只需在搜索关键词后加上 - filetype: pdf 就可以了,而如果只想查pdf 文件,则用 filetype: pdf 就可以指定pdf 文件了。每一个文档在数据库里都有其文件类型,因此,实现它并不困难。

(3)指定范围 Google 可以指定网页的范围,如(all) intitle, (all) intext, (all) inurl, (all) inanchor 等。网页标题只是对HTML 的< title > < /title> 里的内容进行检索,如果 title 的内容与文章真正的标题不一致,则 Google 没有能力检索出来。在 http://www7.scu.edu.au/programme/fullpapers/1921/com1921.htm 里的文章真正的标题是 The Anatomy of a Large- Scale Hypertextual Web Search Engine,而相关 HTML 文件里< title> < /title> 的内容却是 The Anatomy of a Search Engine,因此在 Google 的输入框里键入 allintitle:"The Anatomy of a Large- Scale Hypertextual Web Search Engine",是查不到该文章的。而对文章内容的检索识别是 HTML 文件里的< body > < /body> 中的内容。当然还可以限定时间、指定站点等。通过 site: www.xxx.xxx.xxx 指定站点实现站内搜索是很有用的。

(4)语言问题 Google 支持 66 种语言接口,35 种语言指定检索,Google 对语种的判别不是通过URL 信息(域名中的国别),主要通过 HTML 中的 charset,和对网页内容部分识别的统计信息来联合判断。其实,好多人还把 Google 当词典来用,我们输入"层次分析法 analysis"指定中文网页就可以检索出层次分析法的英文翻译。

(5)同义词检索 选择"搜索所有网站或搜索所有中文网页",输入"计算机",则会把含有"电脑"的网页(不论有没有"计算机")搜索出来,而在简体中文网页里就不会实现这样的功能,因为 Google 采用 Basis Technology 的中文简繁体转换技术。

(6)网页目录 Google 按内容把网页信息分为 15 个大类,主要是通过词频分布与统计技术,对内容进行识别并用分类器(sorter)进行归类。

(7)图像搜索 Google 搜集了 3.9 亿张图像,对图像的描述信息建了一个很大的数据库。图像检索使用的主要是文件名、图片附近的文本、图片的标题以及从图片中提取的部分内容。

(8)相似结果 Google 会省略相似的结果,尤其是来自不同站点页内容相同的文章,有时我们在 Google 显示的结果中不能下载相应的文章时,而省略的结果中可能允许下载,这时 Google 的省略功能就带来了不便。判断内容相似主要可能是根据文章题目、文章大小和部分词语统计信息等。当然还有许多特点与使用技巧,那不是本文讨论的重点。

我们了解了 Google 的技术实现,就可以理解检索过程中出现的种种现象。针对这些特点再去调整检索策略,这样它返回的结果就不再成千上万了。正常情况下,把检索结果控制在百条左右,是一个专业检索水平的标志,这样的结果对我们也才真正的有意义。

参考文献:

1 http://www.google.com/intl/zh-CN/features.html(Accessed Jun. 25, 2003)

2 http://www.google.com/intl/zh-CN/why_use.html(Accessed Jun. 25, 2003)

3 Junghoo Cho, Hector Garcia-Molina and Lawrence Page. Efficient crawling through URL ordering. http://www7.scu.edu.au/programme/fullpapers/1919/com1919.htm (Accessed Jun. 26, 2003)

4 Sergey Brin and Lawrence Page. The Anatomy of a Large-Scale Hypertextual Web Search Engine. http://www-db.stanford.edu/~backrub/google.html(Accessed Jun. 26, 2003)

5 The Google Pagerank Algorithm and How It works. http://www.iprcom.com/papers/pagerank/(Accessed Jun. 26, 2003)

(作者 E-mail: huabolin@sohu.com)