

搜索引擎技术介绍

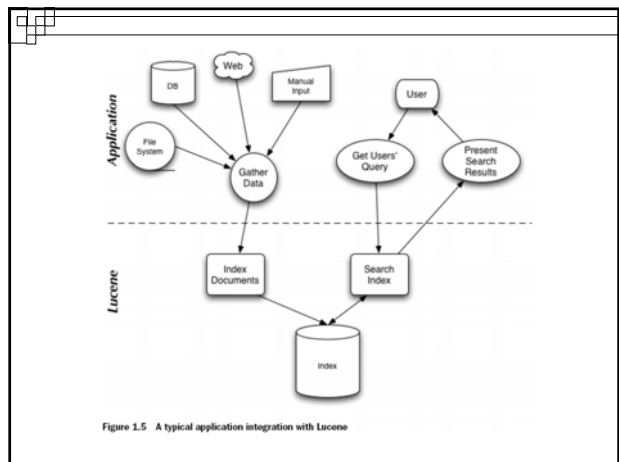
王栋

Topics

- 概述
- 信息检索模型
- 信息检索系统的评价标准
- Web搜索引擎的难点
- Web搜索引擎体系结构
- Web Crawler
- 预处理
- 索引和查找
- 检索结果排序

概述

- 搜索引擎属于信息检索(Information Retrieval, IR)范畴
- 信息检索的基本任务
 - 如何找到并定位特定资源?
 - 这些资源可能来自
 - Web
 - 数据库
 - 文件系统
 -
- 如果目标资源是Web, 就称为Web搜索引擎
 - Google, 百度, Yahoo!



信息检索模型(1/3)

- 信息检索模型 (IR model) 可形式化地表示为一个四元组:

$$\langle D, Q, F, R(q, d) \rangle$$
- 其中D是一个文档集合, Q是一个查询集合, F是一个对文档和查询建模的框架, $R(q, d)$ 是一个排序函数, 它给查询q和文档d之间的相关性赋予一个排序值, 即相关性评价。
- 常见的信息检索模型有:
 - 布尔模型 (Boolean Model)
 - 向量空间模型 (Vector Space Model)
 - 概率模型 (Probabilistic Model)
 - 推理网络模型 (Inference Network Model)

信息检索模型(1/2)

- 信息检索的一个核心问题是如何决定查询和文档之间的相关性, 即信息检索模型中的排序函数 $R(q,d)$ 。
- 常用的相关性评价方法是向量空间模型(Vector Space Model, VSM)
- 向量空间模型基于共有词汇假设 (shared bag of words), 即查询和文档都被认为是所有关键词组成的N维向量, 相关性根据他们在向量空间中的夹角的cosine值表示, 即

$$R(d, q) = \cos(d, q) = \frac{d \cdot q}{|d| \times |q|}$$
- 那么如何决定N维向量每一维的权重, 即N维向量中每个关键词的权重呢? ?

信息检索模型(2/2)

- 根据信息论原理，信息单位出现的频率越大，携带的信息越小。这就是说出现频率很高的词对于文档区分的作用很小，比如汉语中的“的”，英语中的“the”。
- 基于这一原理，“逆文本频率指数”（Inverse Document Frequency, IDF）通常被用来计算关键词的权重。关键词t的IDF值可以被表示为：

$$IDF(t) = \log(N / df(t))$$
 其中N是所有文档总数，df(t)表示单词t的文档频率(Document Frequency)，即单词t在多少篇文章中出现。
- IDF是一个单词在语言中的统计特性，所以少量新文档加入对它影响很小，可以一次计算后作为单词的属性使用。
- 把TF(t, d)定义为单词t在文档d中的出现频率，那么文档d中关键词t的权重可以表示为：

$$Weight(t, d) = TF(t, d) * IDF(t)$$
 其中，IDF(t)对单词t来说是一个全局权值，而TF(t, d)则是单词t在文档d中的局部权值。

原理

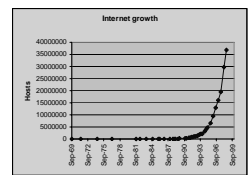
- 根据TF*IDF公式，文档集中包含某一词条的文档越多，说明它区分文档类别属性的能力越低，其权值越小；
- 另一方面，某一文档中某一词条出现的频率越高，说明它区分文档内容属性的能力越强，其权值越大。

信息检索系统的评价标准

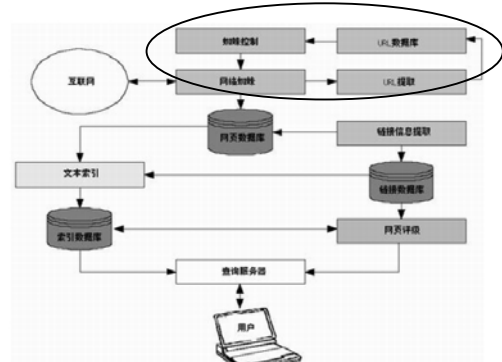
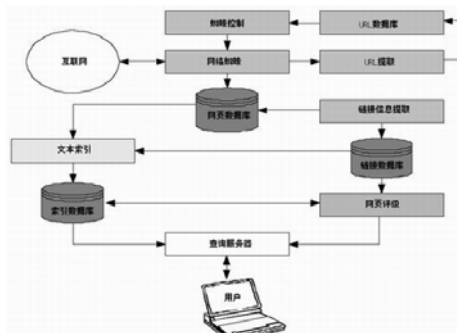
- “效率”几乎是任何计算机系统都需要考虑的问题，比如算法的时空效率，对于信息检索系统，重要的效率指标通常有：
 - 系统的查询响应时间（Response time）
 - 系统的查询吞吐量（Request throughput）。
- “效果”关注用户需求的满足程度，对于信息检索系统通常有两个指标：查全率（Recall）和查准率（Precision）。
 - 查全率定义为检索结果集中的相关文档占整个文档全集中的相关文档的百分比
 - 查准率定义为检索结果集中与用户查询相关的文档占整个检索结果中所有文档的百分比。
 - 查全率是衡量检索系统取回相关信息的能力，查准率是衡量检索系统拒绝非相关信息的能力。实验证明，在信息检索中，查全率和查准率之间存在着相反的相互依赖关系，即查准率和查全率往往不能两全其美，通常查准率高时，查全率低；查全率高时，查准率低。

Web搜索引擎的难点

- 数据
 - 数据规模巨大且增长快
 - 比如，Web上的网页量级是billion，中国的web页面就有几十亿！
 - Web的异构性
 - 多种多样
 - 文本、图片、视频、音频等
 - 非结构化和半结构化数据
 - 比如，文本数据和XML数据
- 用户
 - 如何表达查询需求？
 - 如何解释查询结果？



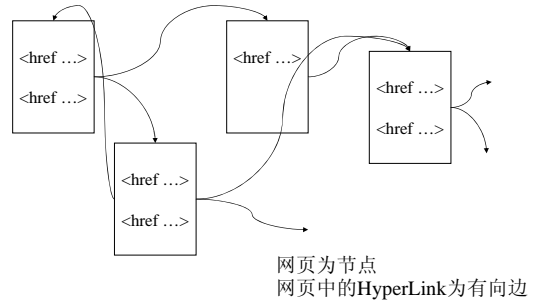
Web搜索引擎体系结构



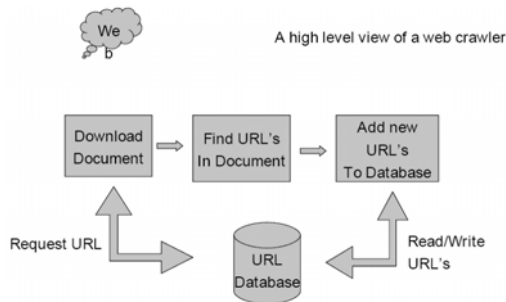
网络爬虫

- Google's mission: *Organize the world's information and make it universally accessible and useful.*
- 第一步要解决信息的获取问题
- 网络爬虫 (Web Crawler) 是搜索引擎的重要组成部分, 它负责把网上的数据抓取 (Crawl) 下来供搜索引擎使用。

Web是一个有向图

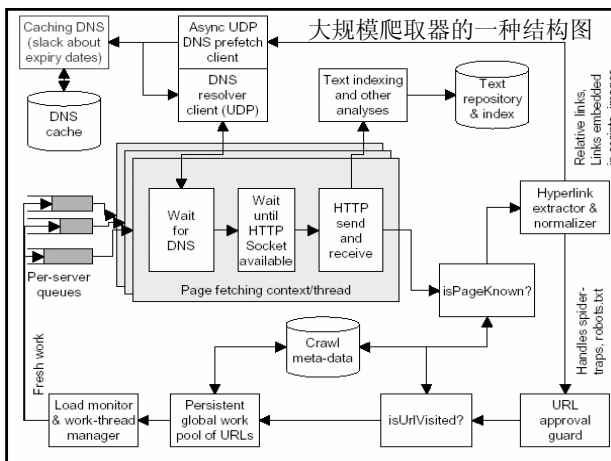


系统框图



High-performance Crawler need...

- Scalable
 - Parallel, distributed
- Fast
 - Bottleneck? Network utilization
- Polite
 - DoS, robot.txt
- Robust
 - Traps, errors, crash recovery
- Continuous
 - Batch or incremental



大规模爬取器：性能和可靠性问题

- 避免让DNS查询成为瓶颈
- 同时并发抓取多个网页 (例如一台机器200个并发)
 - 这是充分利用网络带宽的基础
 - 多进程、多线程
 - 利用异步sockets (Soumen的观点)
 - 用一个数据结构, 显式将一个抓取过程的状态表达出来
 - 检查结束标志
- URL提取中的问题
 - 消除重复, 减少冗余的抓取 (不那么容易, 同义URL问题)
 - 避免"spider traps", 陷入少量网站中

Issue: 消除已经访问过的URL

- 检查某个URL是否已经被抓过了
 - 在将一个新的URL放到工作池之前
 - 要很快，不要在这里形成性能瓶颈（检查将要访问磁盘）
 - 符合条件（即未被访问过）的URLs放到crawler的任务中
- 优化方法
 - 可以通过计算并对比（规格化后的）URL的MD5来实现
 - 利用访问的时空局部性--Cache
 - 高效率的查找表数据结构
 - 用B-树管理
 - Bloom filter
 - 空间效率很高，用于判断某元素是否属于某集合

Diving in the crawlers Take TSE for ex.

陈志杰

预处理

- 对于抓下来的HTML文档，需要解析HTML
 - Word, PDF.....
- 扫描并提取词串
- 英文
 - Stemming: 提取词根
- 中文
 - Segmenting: 分词
- 去掉停用词（Stop Words）
 - "the", "a", etc
 - "的", "地", 等
- 词性标注
- 命名实体识别
 - 日期、数字、机构名、人名等。

中文分词简介(1/3)

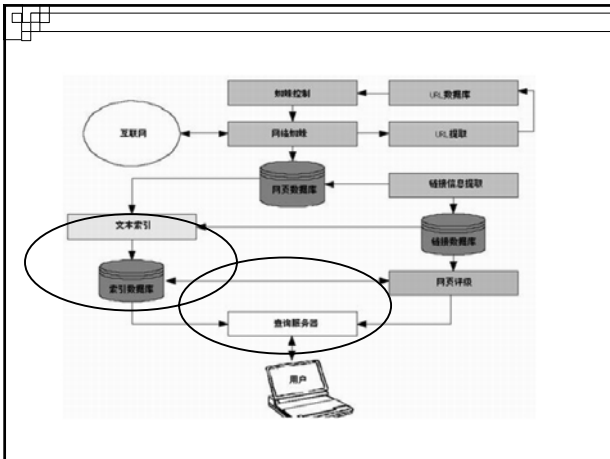
- 因为中文本身存在着很大的歧义性，同样一句话，不同的断句，表达的意思就不一样。这对于计算机去做机器分析，就带来了巨大的困难。
- 下面的中文断句，来自百度广告宣传片：
 - 我知道你不知道我知道你不知道我知道你不知道我知道，你不知道我知道，你不知道我知道。我知道，你不知道我知道，你不知道我知道你，你不知道我知道你，你不知道我知道。你知道你不知道我知道，你知道你不知道我知道你，你不知道我知道你，你不知道我知道你。
- 另外中文的具体含义，还必须放在具体的前后语言环境中去分析。比如：
 - 在慈善**拍卖会**上，世界冠军们夺冠时的「乒乓球**拍卖**完了」
- 中文分词，在具体的算法实现上分为三种：
 - 字符串匹配(正序、逆序、最少切分、最大切分等)
 - 基于理解（词法，句法等方式处理）
 - 基于统计
- 在中文搜索引擎中，目前基本上是这三种算法混合使用。第二种的算法实现起来过于复杂，所以以第一种和第三种算法为主。

中文分词简介(2/3)

- 正向最大匹配法(MM)从左向右匹配词典
- 逆向最大匹配法(RMM)从右向左匹配词典
 - 例子
 - 输入:企业要真正具有用工的自主权
 - MM:企业/要/真正/具有/用工的/自主/权
 - RMM:企业/要/真正/具有/用工的/自/主权
- 全切分
 - 利用统计方法训练得到一个概率模型
 - 比如, $P(\text{人民}|\text{中国}) = 0.6$
 - 根据词典生成各种可能的切分情况
 - 如何枚举? 怎么保存结果?
 - 利用概率模型计算各种切分的可能性, 可能性最大的就是最终结果

中文分词简介(2/3)

- n-gram方法
 - 把单字 (unigram) 或相邻的两个字 (bigram) 或更多看作一个索引项
 - 例子: 全文索引完成
 - unigram (1-gram): 全, 文, 索, 引, 完, 成
 - bigram (2-gram): 全文, 文索, 索引, 引完, 完成
 - 3-gram: 全文索, 文索引, 索引完, 引完成
- 简单, P3实习大家可以考虑bigram分词。



索引和查找

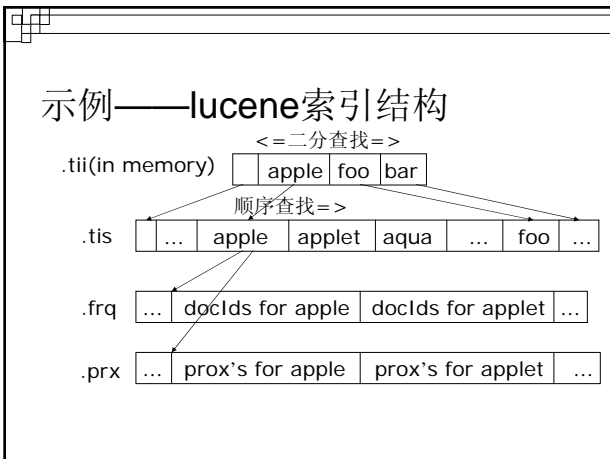
- 两种查找方式
 - 顺序查找
 - 基于索引的查找
- 显然，第一种方式适合对规模小，变化快的数据集查找；第二种方式适合于大规模的静态数据集。
- 现代的数据库系统在查找过程中结合了两种方式。

常见的索引方式

- 后缀数组，倒排索引和Signature files
- 倒排索引(inverted index) 组成
 - 词表 (Dictionary, lexicon)
 - Hash table
 - O(1) lookup
 - complex to expand
 - B-tree
 - O(log n) lookups to find a list
 - easy to expand
 -
 - Postings
 - document ids
 - word positions

倒排索引实现

- 由于词表的规模不会很大，所以在查询时词表通常是常驻内存的。
- 如果文档集很大，那么词的出现位置列表 (word positions) 也可能很大。如果内存不够大可放在外存中。
- 把出现位置列表放在内存中可大大提高查询速度。



Block addressing

- 一种缩小出现位置列表的方法：把文档分成若干个块，在出现位置列表中只记录词出现在哪一块中，而不记录具体位置。再从这一个块中进行顺序查找。这种方式称为Block addressing。
- 例如：

Block 1	Block 2	Block 3	Block 4
This is a text.	A text has many	words. Words are	made from letters.

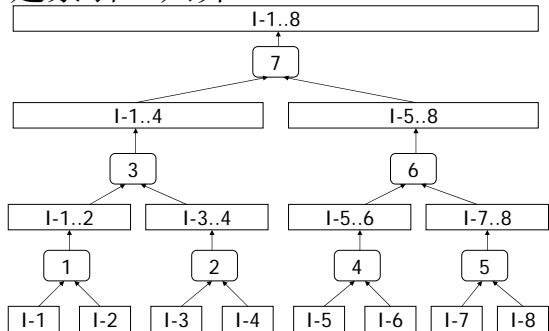
建索引(1/2)

- 1) 抽取posting
 - 文档->词 自然关系的倒置过程, 生成 词->文档
 - 把单词和对应的文档编号, 出现位置相结合, 生成 <word_id, doc_id, pos>三元组 (posting)。
- 2) 排序
 - 先按单词 (字典顺序), 其次文档id, 最后出现位置 pos, 对所有posting排序, 产生倒排表。
- 3) 输出
 - 按顺序将倒排表写到磁盘上。

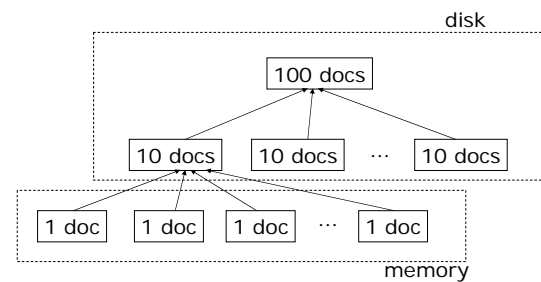
建索引(2/2)

- 对于较小的文档集, 可在完全在内存中对其建立倒排索引, 再写到文件中。
- 当文档集很大时, 问题就出现了: 不可能在内存中对所有的文档建立倒排索引。于是, 必须把文档集分成若干块, 分块建立索引。生成多个倒排文件后, 再把它们归并成一个大文件。
- 如下图所示:

建索引: 归并



一种增量索引算法实现



查询倒排文件

- 从倒排文件中查找分三个基本步骤:
 - 词表的查找。注意每个查询可能包含若干个词。
 - 获得各个词的Postings
 - 对Postings的处理。如处理词的相邻位置关系, 布尔查询等。

实现优化

- I/O系统实现特点:
 - 无缓冲的底层I/O接口效率较好 vs. <stdio.h>
 - 使用内存文件映射或者直接磁盘访问避免多次内存拷贝问题, 从而提高系统效率。
- 索引压缩
- 重要索引词单独索引(天网)

索引压缩

- 索引压缩的目的: 减小索引占用的磁盘空间
 - 磁盘访问是检索的主要开销
 - CPU和磁盘剪刀差恶化
 - CPU时间和IO时间的权衡
- 两个矛盾的目标
 - 高压缩比
 - 低时间开销

Delta encoding

- 存储相邻数据的差, 缩小数据动态范围
 - $a, b, c \dots \Rightarrow a, b-a, c-b \dots$
- docID
 - 包含某个词的docid在.frq中是一个递增序列
0, 2, 3, 5, 6, 10...
 - 对于高频词, 相邻的docid数字接近
 - Delta编码:
0, 2, 1, 2, 1, 4...

Delta encoding

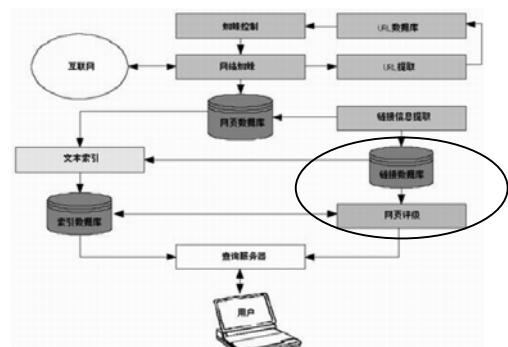
- 词
 - 词在词表中按字典序存放
apple, applet, application, banana ...
 - 相邻词有重复部分
 - 保存差异部分的字符串+相同部分的长度
 - Delta编码:
<0, apple>, <4, t>, <4, ication>,
<0, banana>

结果排序

- 基于统计
 - 根据前述的TF*IDF模型
- 考虑拓扑结构
 - 根据Web页面的链接关系
- 其他
 - 百度竞价排名等

动机

- 一个短查询(1-3个词)可能返回上万个网页甚至更多
- 对返回页面进行排序, 使有用的页面出现在返回结果的顶部



考虑拓扑结构

- 两个常用的web页面排序算法
 - HITS
 - 在轴页面(hub page)和权威页面(authority page)之间互相加强(mutual reinforcement)
 - PageRank
 - 超链权重的传播
 - 基于随机行走模型(Web surfing based on a random walk models)

PageRank and HITS

冯熙铉

信息检索与数据管理

- 信息检索经常和数据管理技术（比如数据库）的研究交叉在一起，但二者是有区别的。
- 数据管理技术处理的是结构化信息，用户的每个操作执行的结果是确定的。
- 信息检索处理的信息包罗万象，除了结构化信息，也可以是非结构化和半结构化信息，这需要信息检索系统具有理解自然语言的能力，而自然语言是不精确的、模糊的、具有二义性的，因而信息检索对用户的查询返回的结果也往往是不精确的。

加强对语义的理解

- 下一代搜索引擎的开发者们认为他们的技术将会能够“理解”那些通过语义而提出的各种问题，
- 和披头士有关的音乐家有哪些？
- 全球最好的大学是哪家？
- 语义搜索引擎
 - Hakia: <http://www.hakia.com/>


Bunny & Rabbit


Tree Structure & Word Similarity
张策

推荐阅读

- 李晓明, 闫宏飞, 王继民: 《搜索引擎-原理, 技术与系统》
- JUSTIN ZOBEL, ALISTAIR MOFFAT, *Inverted Files for Text Search Engines*

Index/IR toolkits

 **Lemur** ■ The Lemur Toolkit for Language Modeling and Information Retrieval
<http://www.lemurproject.org/>

 **Lucene**
<http://lucene.apache.org/nutch/>

- Java-based indexing and search technology, provide web search application software

End

中文切词

- 中文处理
 - 可以选用Bigram索引：中文大部分词是两个词
 - 比如：全文索引 => 全文，文索，索引
- 英文处理
 - Stemming不是必须的，选作。
 - <http://tartarus.org/martin/PorterStemmer/>
- 中文和英文在词表的设计上是不同的，可以分开处理，也可以统一按中文处理。