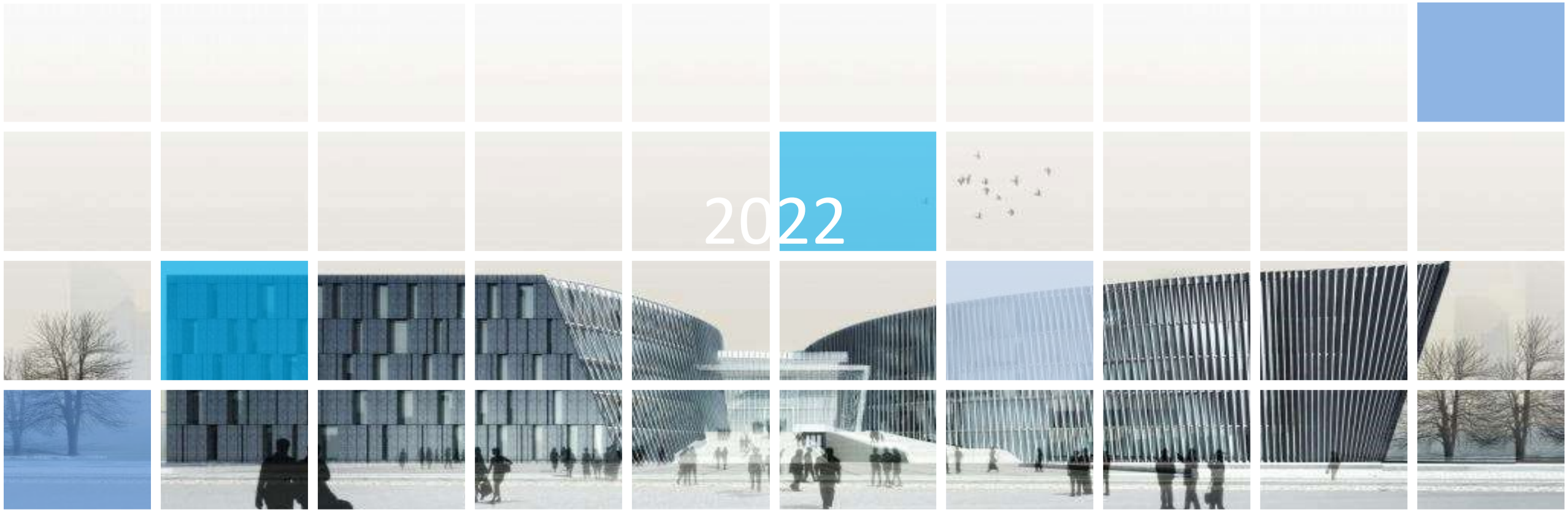


南京大学信息管理学院

信息检索

邓三鸿

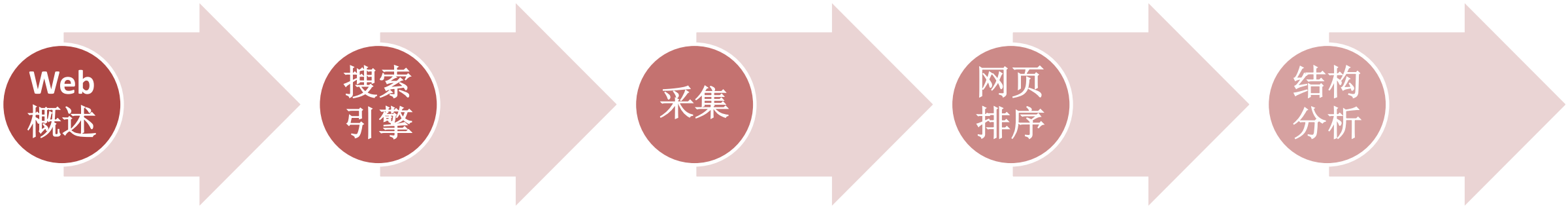
njuir@sina.com

PART Eight

Web信息采集与搜索引擎

Search Engine

Web信息处理概要



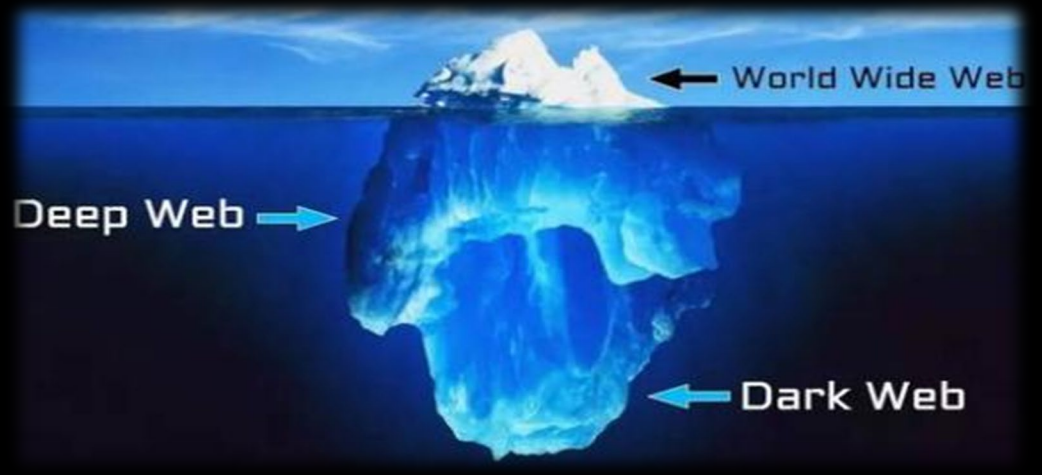
Internet与Web

- Internet
  - 因特网，国际互联网
  - 1969秋，ARPAnet，1972年国际联网
  - TCP/IP

- World Wide Web
  - 万维网
  - 1999.12，Tim命名WWW
  - Http

Tim Berners-Lee（1955-）

暗网与深网



Darknet或Dark Web：需要通过特殊软件、特殊授权、或对电脑做特殊设置才能连上的网络；

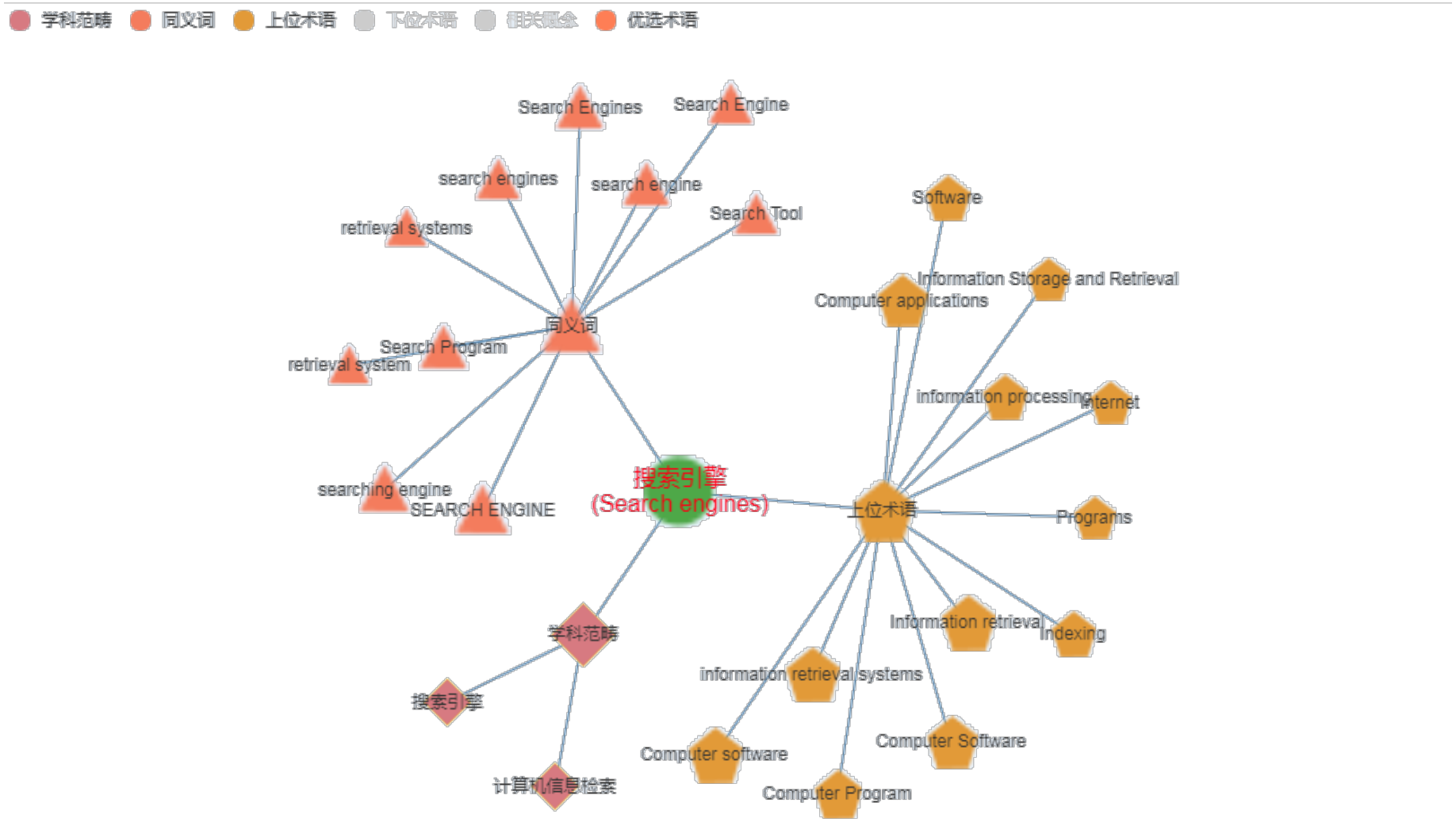
Deep Web：互联网上那些不能被标准搜索引擎索引的非表面网络内容。

暗网是深网的一个子集。

搜索引擎

- Search Engine
- 根据一定的策略、运用特定的计算机程序从互联网上搜集信息，在对信息进行组织和处理后，为用户提供检索服务，将用户检索相关的信息展示给用户的系统。

SE的相关研究领域



智能拓展

数据来自万方数据知识服务平台

2018.12

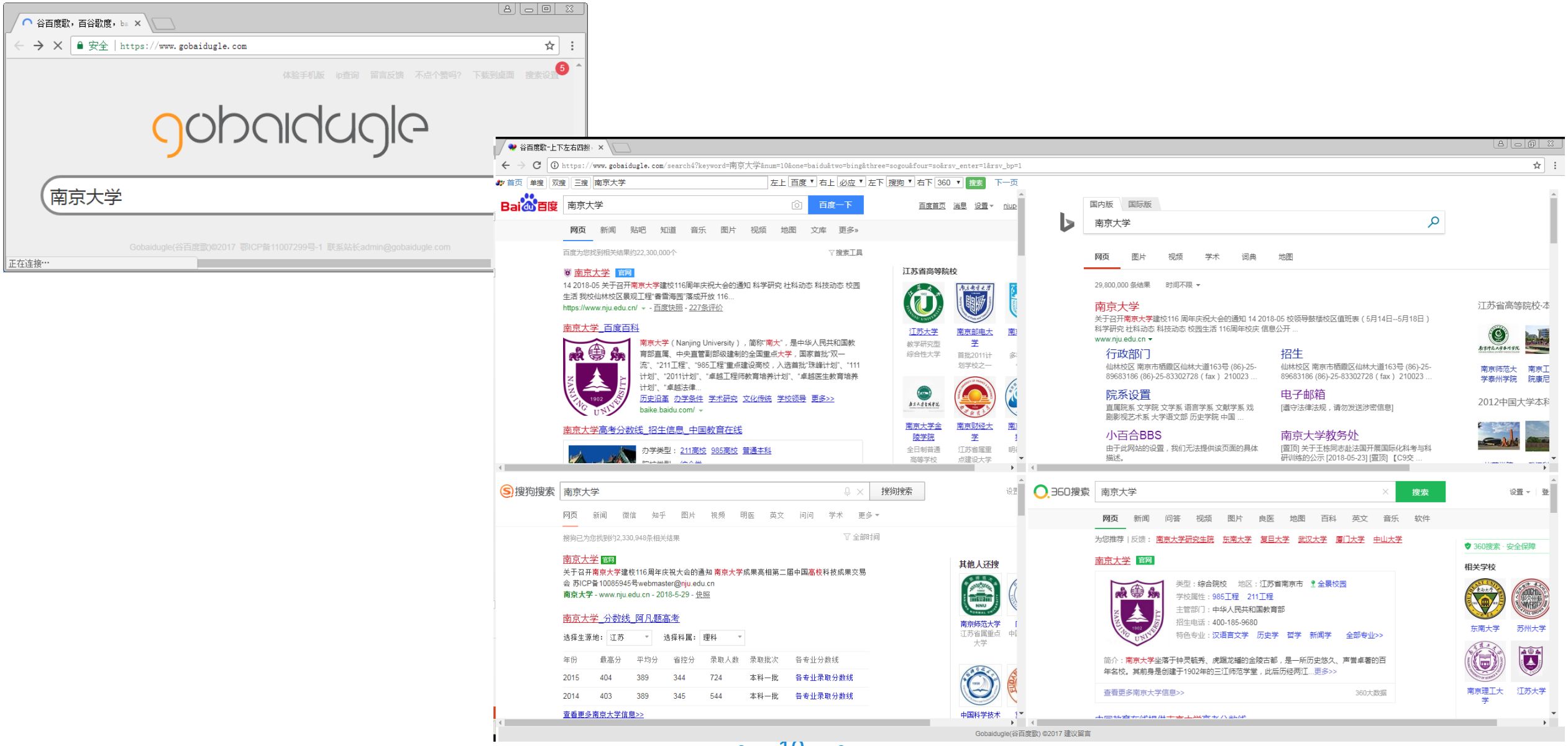
搜索引擎的发展

- 起源：FTP文件搜索
  - （以Archie为代表，1990-）
- 第一代搜索引擎：分类目录
  - （以雅虎为代表，1995-）
- 第二代搜索引擎：关键词搜索引擎
  - （以Google为代表，1998-）
- 第三代搜索引擎：智能搜索引擎（自然语言）
  - （发展中，如ask.com）

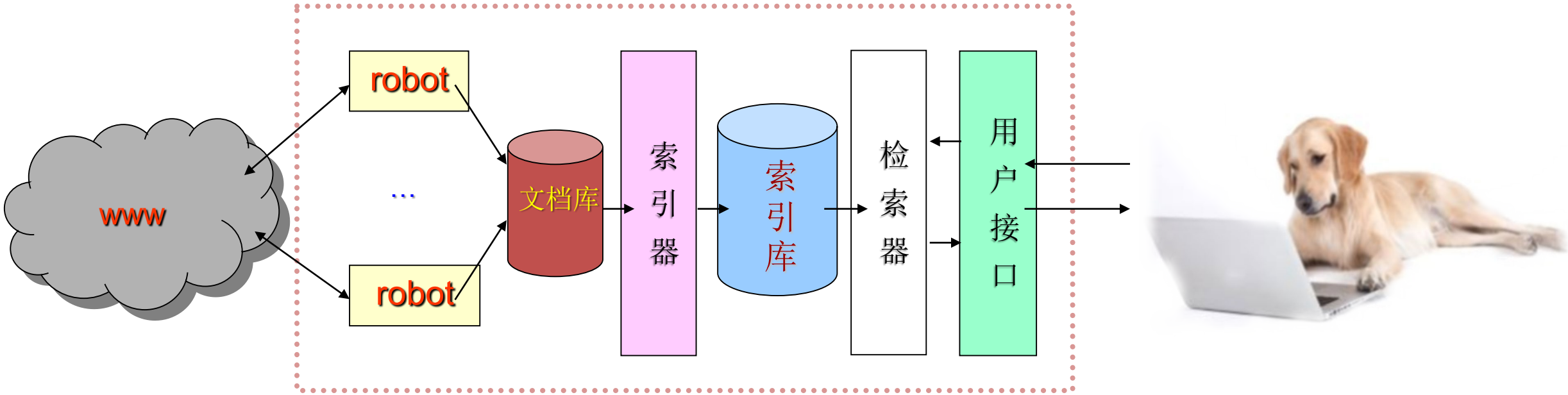
搜索引擎的分类

- **根据检索方式分类**
  - 分类目录、关键词搜索引擎、混合搜索引擎
- **根据信息覆盖范围及适用用户群分类**
  - 综合搜索引擎、专用搜索引擎（垂直搜索引擎）
- **根据搜索范围分类**
  - 独立搜索引擎、集成搜索引擎/元搜索引擎

集成/元搜索引擎


谷百度歌，百谷歌度，
安全 https://www.gobaidugle.com
体验手机版 ip查询 留言反馈 不点个赞吗? 下载到桌面 搜索设置
gobaidugle
南京大学
Gobaidugle(谷百度歌)©2017 鄂ICP备11007299号-1 联系站长admin@gobaidugle.com
正在连接…
谷百度歌-上下左右四指
https://www.gobaidugle.com/search4?keyword=南京大学&num=10&one=baidu&two=bing&three=sogou&four=so&rsv_enter=1&rsv_bp=1
首页 单搜 双搜 三搜 南京大学 左上 百度 右上 必应 左下 搜狗 右下 360 搜索 下一页
百度 南京大学 百度一下
网页 新闻 贴吧 知道 音乐 图片 视频 地图 文库 更多»
南京大学 官网
南京大学_百度百科
南京大学高考分数线_招生信息_中国教育在线
江苏省高等院校
江苏大学
南京邮电大学
南京大学金陵学院
南京财经大学
国内版 国际版
南京大学
网页 图片 视频 学术 词典 地图
南京大学
行政部门
招生
院系设置
电子邮箱
小百合BBS
南京大学教务处
搜狗搜索 南京大学 搜狗搜索
网页 新闻 微信 知乎 图片 视频 明医 英文 问问 学术 更多
南京大学 官网
南京大学_分数线_阿凡题高考
选择生源地：江苏 选择科属：理科
年份 最高分 平均分 省控分 录取人数 录取批次 各专业分数线
2015 404 389 344 724 本科一批 各专业录取分数线
2014 403 389 345 544 本科一批 各专业录取分数线
查看更多南京大学信息>>
其他人还搜
南京师范大学
中国科学技术
360搜索 南京大学 搜索
网页 新闻 问答 视频 图片 良医 地图 百科 英文 音乐 软件
南京大学 官网
类型：综合院校 地区：江苏省南京市
学校属性：985工程 211工程
主管部门：中华人民共和国教育部
招生电话：400-185-9680
特色专业：汉语言文学 历史学 哲学 新闻学
查看更多南京大学信息>>
360搜索 · 安全保障
相关学校
东南大学
苏州大学
南京理工大学
江苏大学
Gobaidugle(谷百度歌) ©2017 建议留言

搜索引擎的基本结构

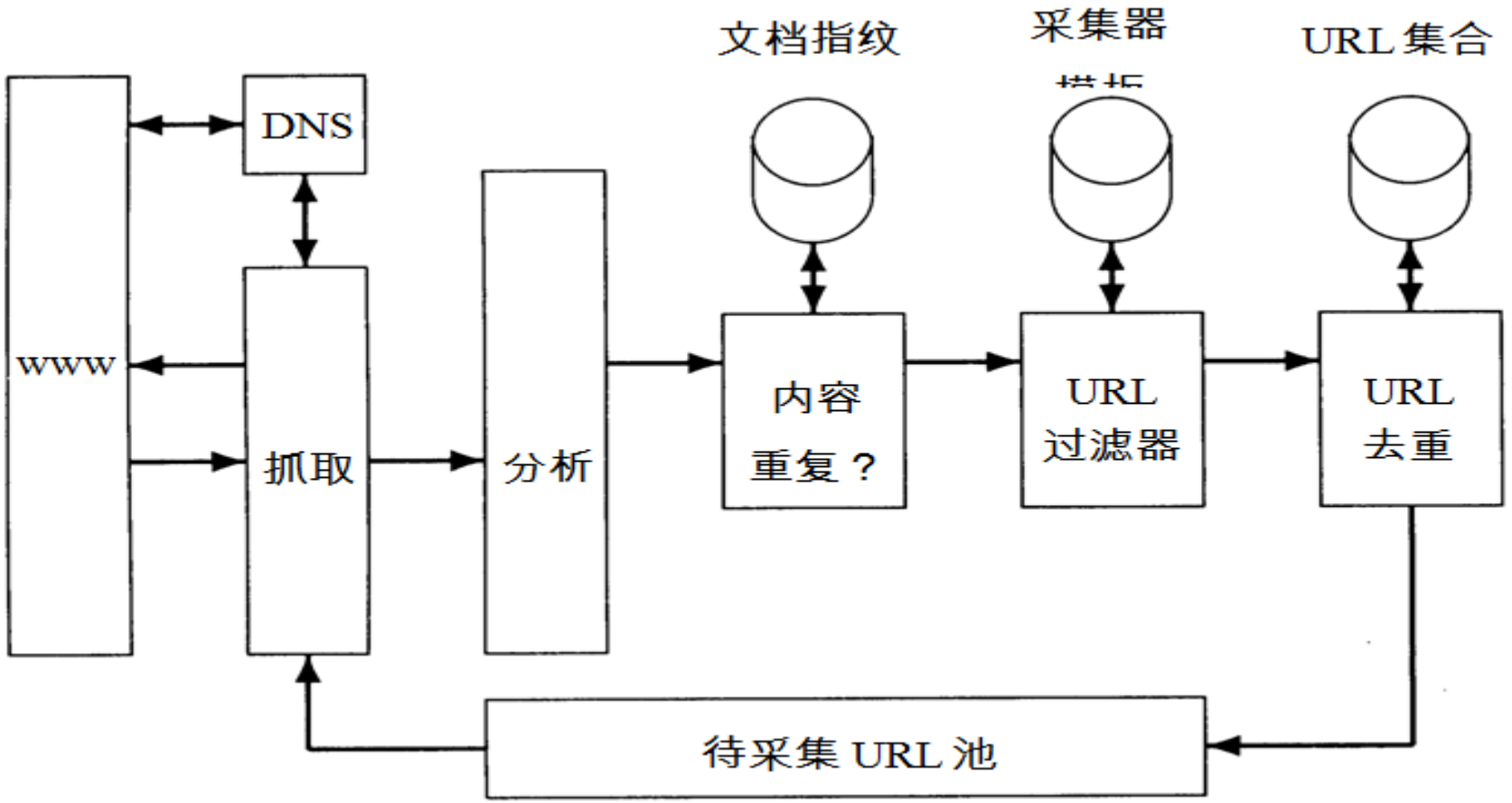


采集 处理 服务

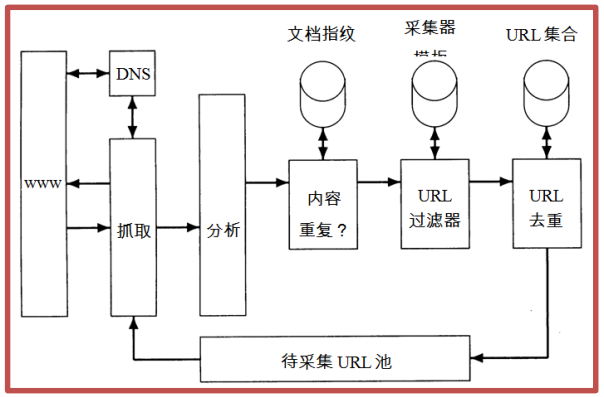
基本的采集过程

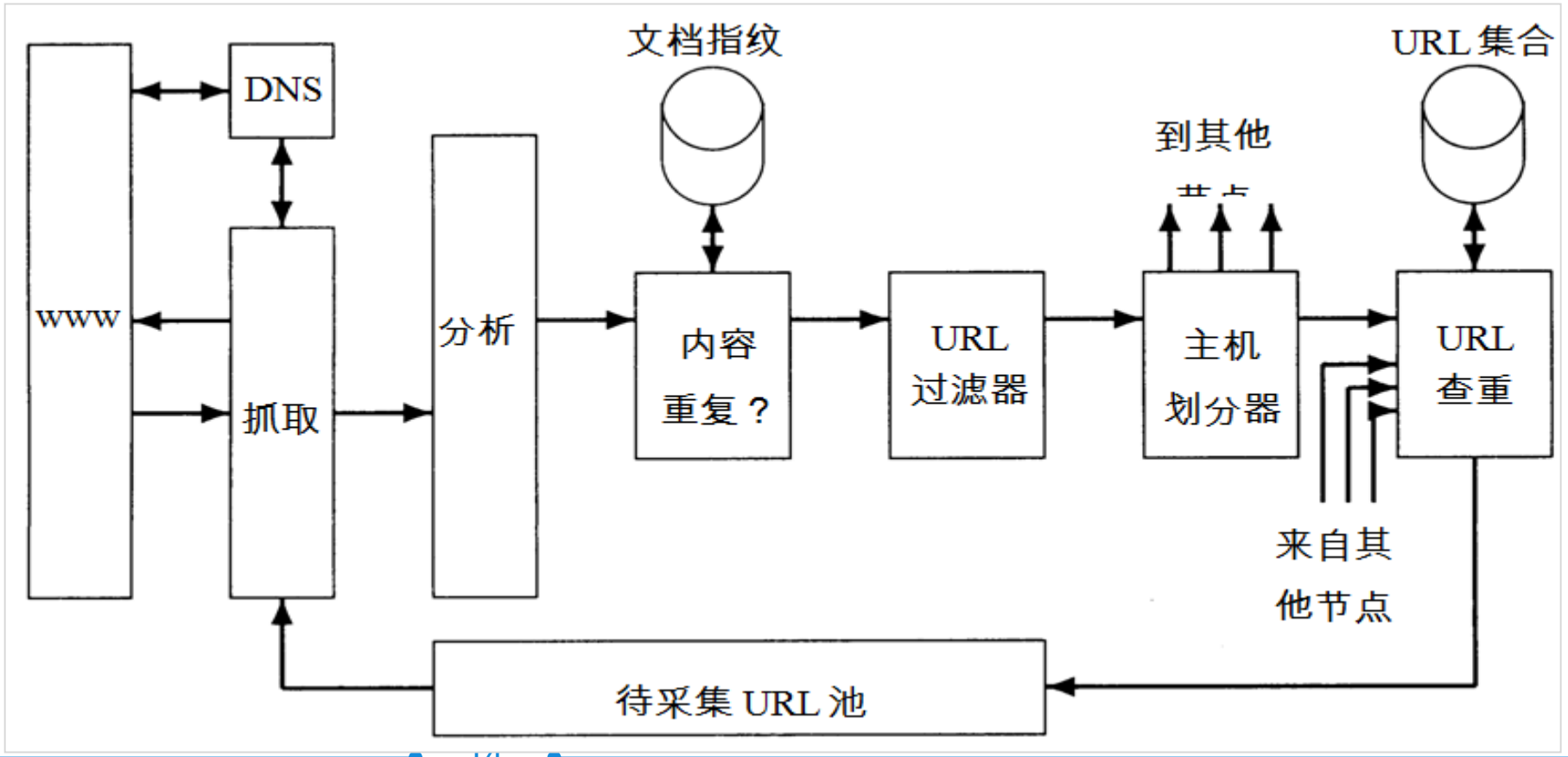
- 初始化采集URL种子队列；
- 重复如下过程：
  - *从队列中取出URL*
  - *下载并分析网页*
  - *从网页中抽取更多的URL*
  - *将这些URL放到队列中*
- 基本假设：**Web的连通性很好**

单机采集架构



分布式采集器


文档指纹
采集器
URL集合
DNS
www
抓取
分析
内容
重复？
URL
过滤器
URL
去重
待采集 URL 池


文档指纹
URL集合
DNS
到其他
www
抓取
分析
内容
重复？
URL
过滤器
主机
划分器
URL
查重
来自其
他节点
待采集 URL 池

爬行策略


定点策略


定题策略

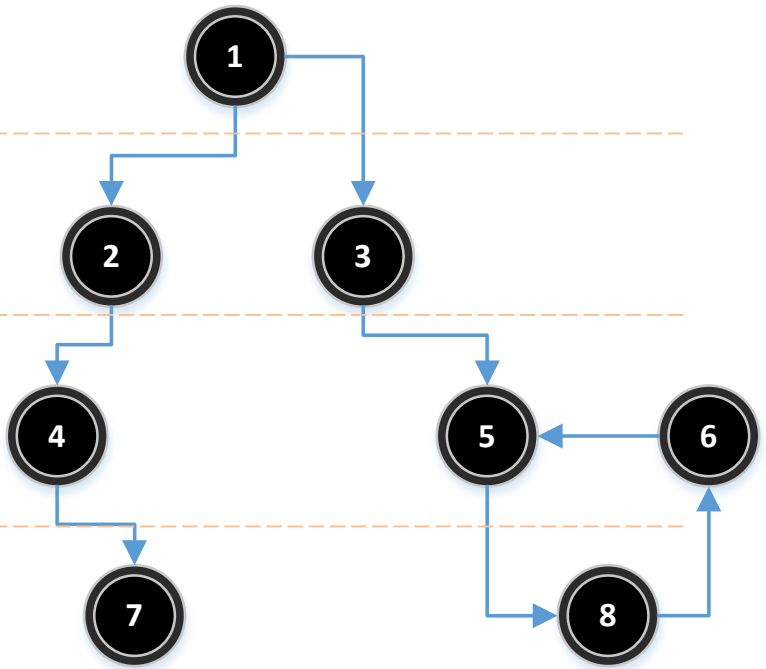

广度优先


深度优先


大站/要站优先


Partial PageRank/OPIC排序

广度优先与深度优先



Breadth-First Search，即广度优先搜索，又称作宽度优先搜索，或横向优先搜索，简称**BFS**

1-2-3-4-5-7-8-6

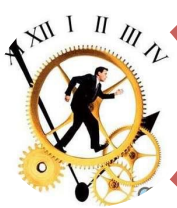
Depth-First Search，即深度优先搜索，简称**DFS**

1-2-4-7-3-5-8-6

采集时机

- **即时抓取**
  - 用户提交查询的时候即时去网上抓取网页
  - 缺点：系统效益不高（重复抓取网页）
- **预先搜集（直接或间接）**
  - 定期搜集
    - 每次搜集替换上一次的内容
    - 优点：实现简单
    - 缺点：时新性（freshness）不高；重复搜集带来的额外宽带开销
- **增量搜集**
  - 开始时搜集一批网页，以后
    - 只搜集新出现的网页
    - 搜集那些在上次搜集后有过改变的网页
    - 发现自从上次搜索后已经不再存在了的网页，并从网页库中删除
  - 优点：每次搜集的网页量不是很大，可以经常启动搜集过程；时新性比较高
  - 缺点：系统实现比较复杂；不仅搜集过程复杂，而且后续创建索引的过程也很复杂

注意事项



效率

如何利用尽量少的资源（计算机设备、网络带宽、时间）来完成预定的网页搜集量

礼节

网页被搜索引擎索引，从而可能得到更多的访问流量
搜索引擎的“密集”抓取活动阻碍了用户通过浏览器的访问

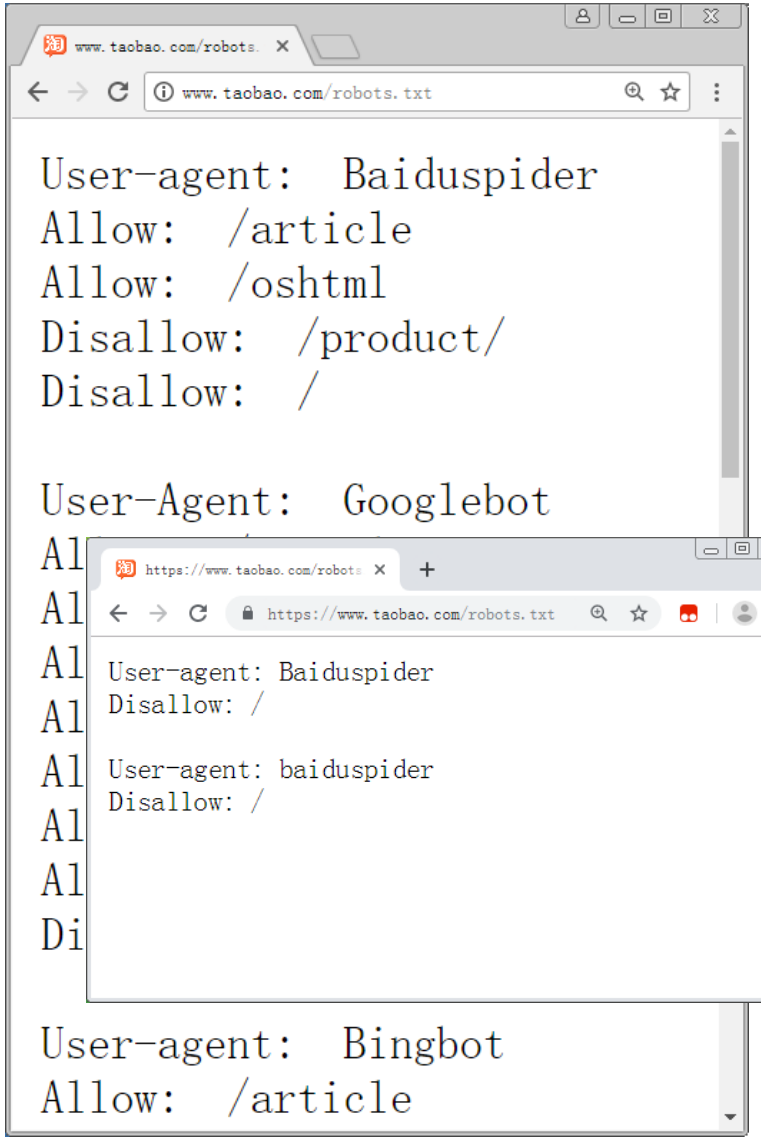
质量

在有限的时间，搜集有限的网页，不要漏掉那些很重要的网页
保证每个网页不被重复抓取

Robots协议

- 网站通过该协议告诉搜索引擎哪些页面可以抓取，哪些页面不能抓取。
  - ✓ User-agent　　*User-agent: Baiduspider*
  - ✓ Disallow　　*Disallow: /cgi-bin/*.htm*
  - ✓ Allow　　*Allow: .gif$*
  - ✓ Sitemap

- **meta Robots标签**
  - ✓ *<meta name="robots" content="all" />*
  - ✓ name可以针对需要进行修改
  - ✓ content的4个指令：index、noindex、follow、nofollow
    - index 指令告诉搜索机器人抓取该页面；
    - follow 指令表示可以沿着该页面上的链接继续抓取下去；



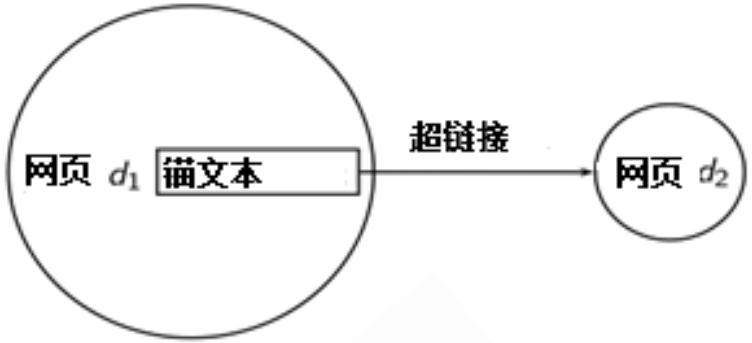
几个概念

- **锚文本**
  - 把关键词做一个链接，指向别的网页，这种形式的链接（文本）就叫作锚文本
- **入链**
  - In-link/inbound link
- **出链**
  - Out-link/outbound link
- **中心节点**
  - Hub
- **权威节点**
  - Authority

网页排序与链接分析

- 早期搜索引擎主要是比较查询与页面的相关度
  - TF-IDF、SVM、Cosine……
- 链接分析，源于对Web结构中超链接的多维分析。
- 类似于引文分析
  - 论文的价值可以用引用频次来衡量
- 竞价排名？！

Web是一个有向图



假设1: 超链接代表了某种质量认可信号

- 超链 $d_1 \rightarrow d_2$ 表示 d_1的作者认可 d_2 的质量和相关性

假设 2: 锚文本描述了文档 d_2 的内容

- 这里的锚文本定义比较宽泛，包括链接周围的文本

- 例子： “You can find cheap cars <a href =http://…>*here* . ”
- 锚文本: “You can find cheap cars here”

链接中心

iSchools

University of Pittsburgh School of Information Sciences

锚文本的价值

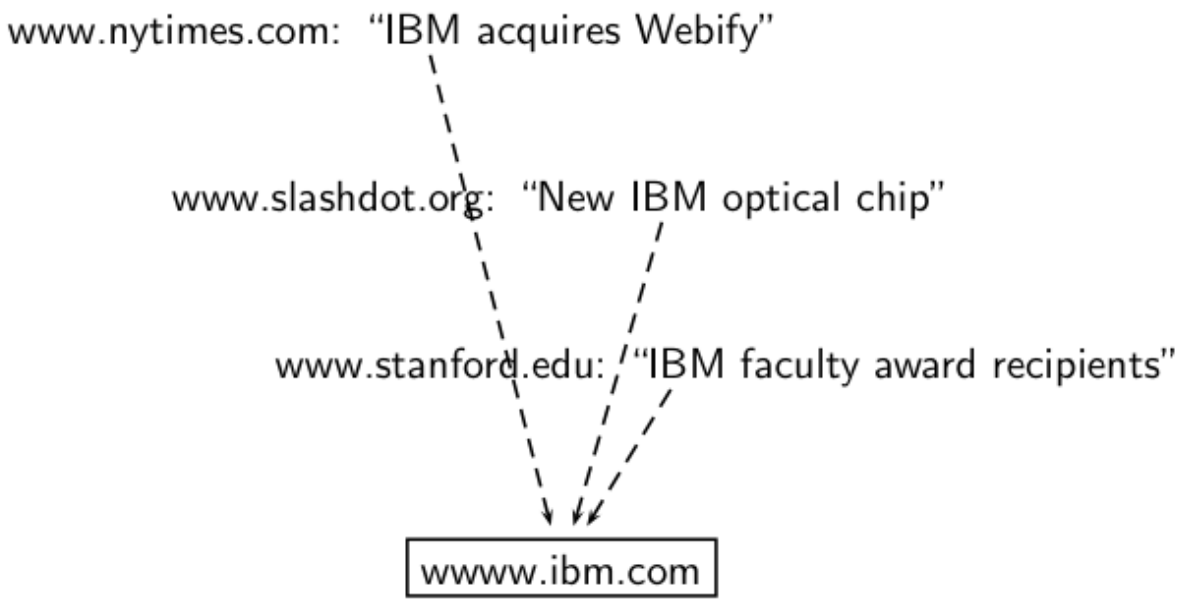
- 后者往往效果好于前者

[d_2中文本] vs. [d_2中文本] + [锚文本 → d_2]

- 例子: 查询 *IBM*
  - IBM 的版权页匹配上
  - 很多作弊网页匹配上
  - IBM的wikipedia页面
  - 可能与IBM 的主页并不匹配!
  - … 也许 IBM 的主页上大部分都是图

- 而按照 [锚文本 → d_2] 来搜索效果会比较好
  - 这种表示下，出现IBM最多的是其主页 www.ibm.com

锚文本指向示例



- 锚文本往往比网页本身更能揭示网页的内容
- 在计算过程中，锚文本应该被赋予比文档中文本更高的权重

引用分析

- 引用分析：科技文献中的引用分析
- 一个引用的例子: “Miller (2001) has shown that physical activity alters the metabolism of estrogens.”
- 可以把“Miller (2001)” 看成是两片学术文献之间的超链接
- 在科技文献领域使用这些“超链接”的一个应用:
  - 根据他人引用的重合率来度量两篇文献的相似度，这称为共引相似度
  - 在Web上也存在共引相似度：Google中提供的 “find pages like this” 或者 “Similar” 功能

引用分析

- 另一个应用:引用频率可以用度量一篇文档的影响度
  - 最简单的度量指标：每篇文档都看成一个投票单位，引用可以看成是投票，然后计算一篇文档被投票的票数。当然这种方法不太精确。
- 在Web上： **引用频率=入链数**
  - 入链数目大并不一定意味着高质量...
  - ...主要原因是因为存在大量作弊链接…
- 更好的度量方法： 对不同网页来的引用频率进行加权
  - 一篇文档的投票权重来自于它本身的引用因子
  - 会不会出现循环计算？答案是否定的，实际上可以采用良好的形式化定义

PageRank

Larry Page（1973.3-）

PageRank

1998.1申请，2001.9授权，US专利号：6,285,999

Google排名的重要组成部分

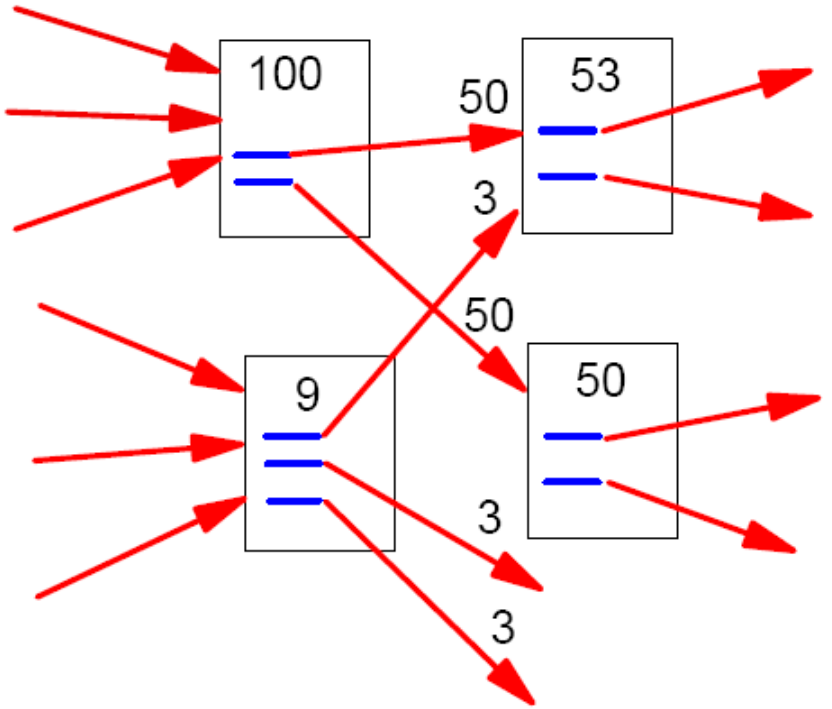
主要思想

拥有**越多、越重要入链**的页面越有价值

原始的PageRank公式

$$R(u) = c \sum_{v \in B_u} \frac{R(v)}{N_v}$$

$R(u)$和$R(v)$是分别是网页u、v的PageRank值，B_u指的是**指向**网页u的网页集合、N_v是网页v的**出链**数目。



一个网页的PageRank等于所有的指向它的网页的PageRank的分量之和(c为归一化参数)。网页的每条出链上每个分量上承载了相同的PageRank分量。

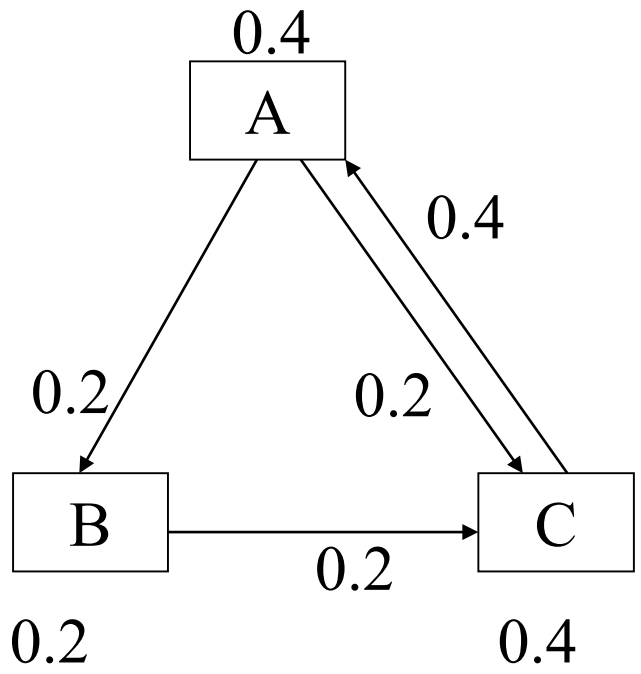
PageRank的特点

(1)一个网页如果它的入链越多，那么它也越重要(PageRank越高)；

(2)一个网页如果被越重要的网页所指向，那么它也越重要(PageRank越高）。

类比：（1）打电话；（2)微博粉丝

简单计算的例子(c=1)



R(A)=R(C)

R(B)=0.5R(A)

R(C)=R(B)+0.5R(A)

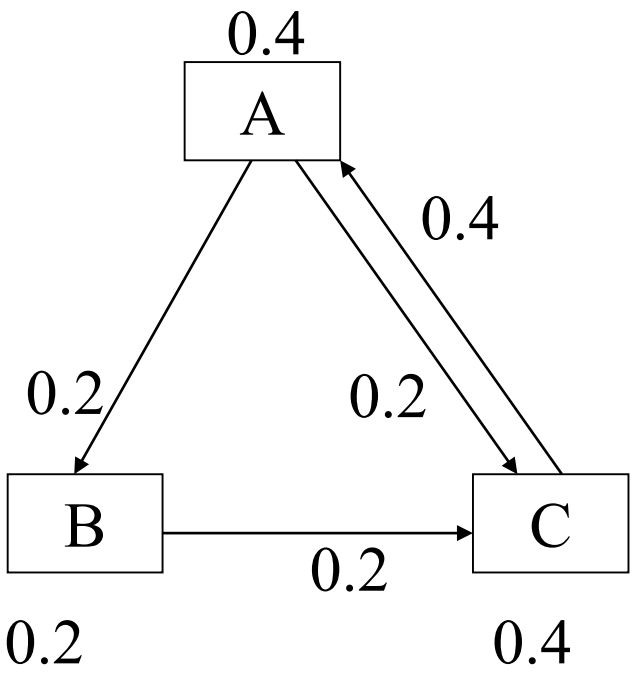
R(A)+R(B)+R(C)=1

解上述方程得:

R(A)=R(C)=0.4

R(B)=0.2

简单计算的例子(c=1)：迭代法求解



$R(A)=R(C)$

$R(B)=0.5R(A)$

$R(C)=R(B)+0.5R(A)$

$R(A)+R(B)+R(C)=1$

迭代次数	R(A)	R(B)	R(C)
0	1/3	1/3	1/3
1	1/3	1/6	1/2
2	1/2	1/6	1/3
3	1/3	1/4	5/12
…	…	…	…
收敛	2/5	1/5	2/5

转化成矩阵形式

- 令$\boldsymbol{R}$表示所有N个网页的PageRank组成的列向量，令网页间的连接矩阵$\boldsymbol{L}=\{l_{ij}\}$，$P_i$有链接指向$P_j$时，$l_{ij}=1$，否则$l_{ij}=0$。对$\boldsymbol{L}$的每行进行归一化，即用$P_i$的出度$N_i$去除得到矩阵$\boldsymbol{A}=\{a_{ij}\}$，$a_{ij}=l_{ij}/N_i$，则有($\boldsymbol{A}^{\mathrm{T}}$表示$\boldsymbol{A}$的转置矩阵)：

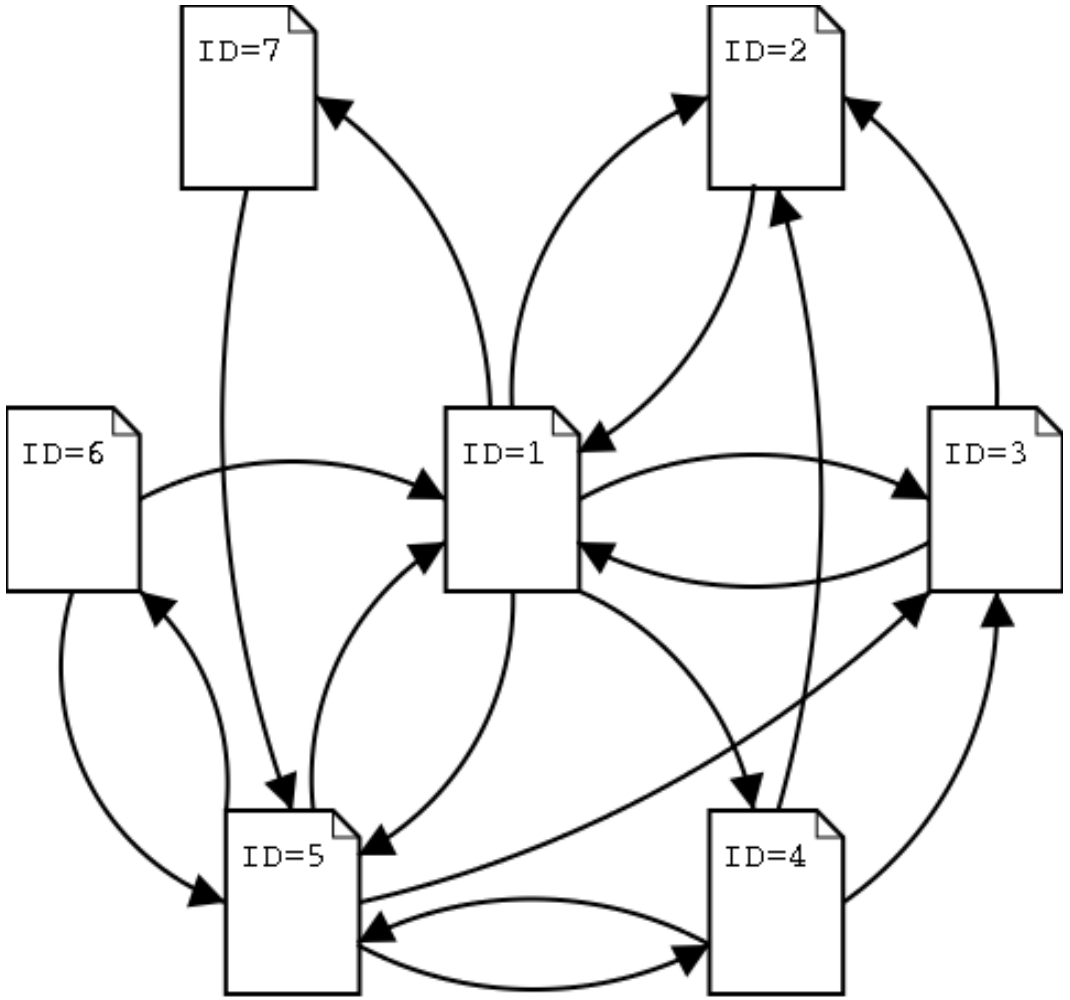
$$\boldsymbol{R}=c\boldsymbol{A}^{\mathrm{T}}\boldsymbol{R} \iff c^{-1}\boldsymbol{R}=\boldsymbol{A}^{\mathrm{T}}\boldsymbol{R}$$

根据线性代数中有关特征向量和特征值的理论，R是矩阵A^{T}的c^{-1}特征值对应的特征向量

$$\begin{aligned} R(A)&=R(C)\\ R(B)&=0.5R(A)\\ R(C)&=R(B)+0.5R(A) \end{aligned}$$

$$\begin{bmatrix} R(A)\\ R(B)\\ R(C) \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1\\ 0.5 & 0 & 0\\ 0.5 & 1 & 0 \end{bmatrix} \begin{bmatrix} R(A)\\ R(B)\\ R(C) \end{bmatrix}$$

一个稍微复杂的例子



Page ID	OutLinks
1	2,3,4,5,7
2	1
3	1,2
4	2.3.5
5	1,3,4,6
6	1.5
7	5

$$A=\begin{pmatrix} 0&1&1&1&1&0&1\\ 1&0&0&0&0&0&0\\ 1&1&0&0&0&0&0\\ 0&1&1&0&1&0&0\\ 1&0&1&1&0&1&0\\ 1&0&0&0&1&0&0\\ 0&0&0&0&1&0&0 \end{pmatrix}$$

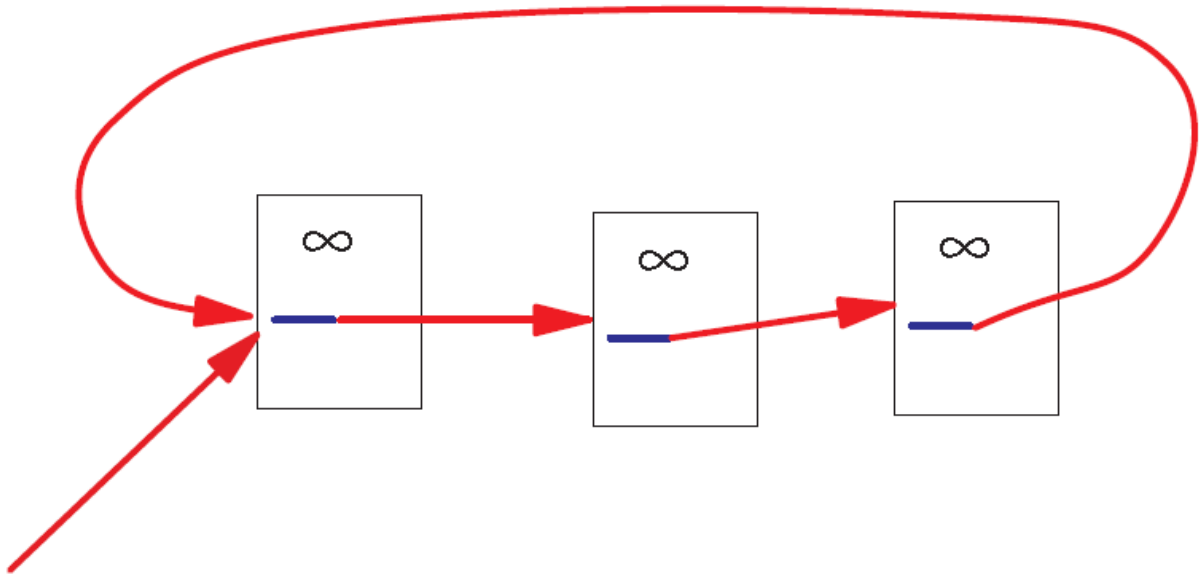
计算过程

则归一化后A = $\begin{pmatrix} 0 & 1/5 & 1/5 & 1/5 & 1/5 & 0 & 1/5 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1/3 & 1/3 & 0 & 1/3 & 0 & 0 \\ 1/4 & 0 & 1/4 & 1/4 & 0 & 1/4 & 0 \\ 1/2 & 0 & 0 & 0 & 1/2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$ $R = cA^{T}R$，令c=1，解得

R = $\begin{pmatrix} 0.69946 \\ 0.38286 \\ 0.32396 \\ 0.24297 \\ 0.41231 \\ 0.10308 \\ 0.13989 \end{pmatrix}$ Normalized = $\begin{pmatrix} 0.303514 \\ 0.166134 \\ 0.140575 \\ 0.105431 \\ 0.178914 \\ 0.044728 \\ 0.060703 \end{pmatrix}$

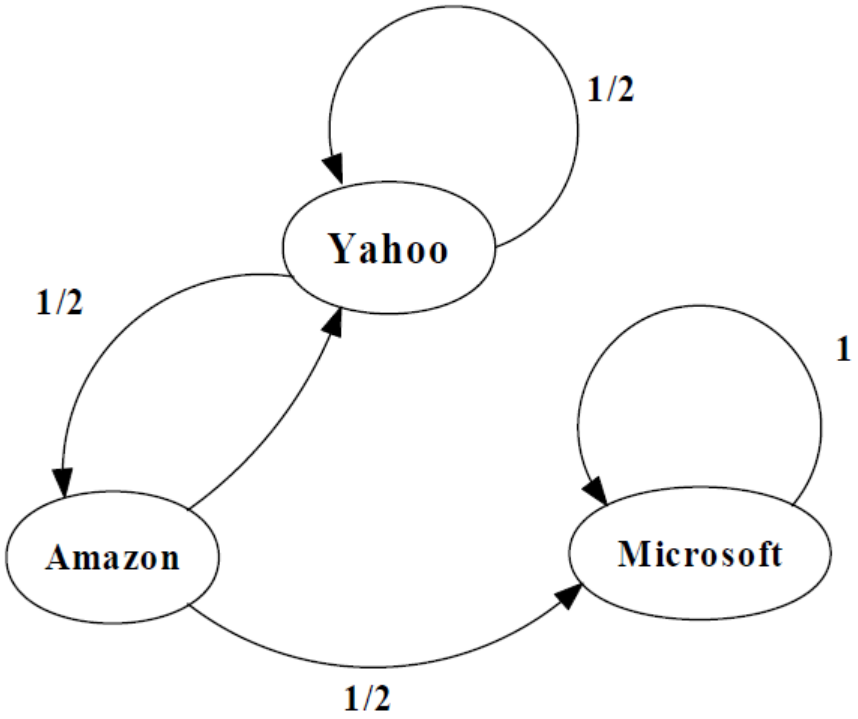
原始PageRank的一个不足

A loop:



图中存在一个循环通路，每次迭代，该循环通路中的每个节点的PageRank不断增加，但是它们并不指出去，即不将PageRank分配给其他节点！

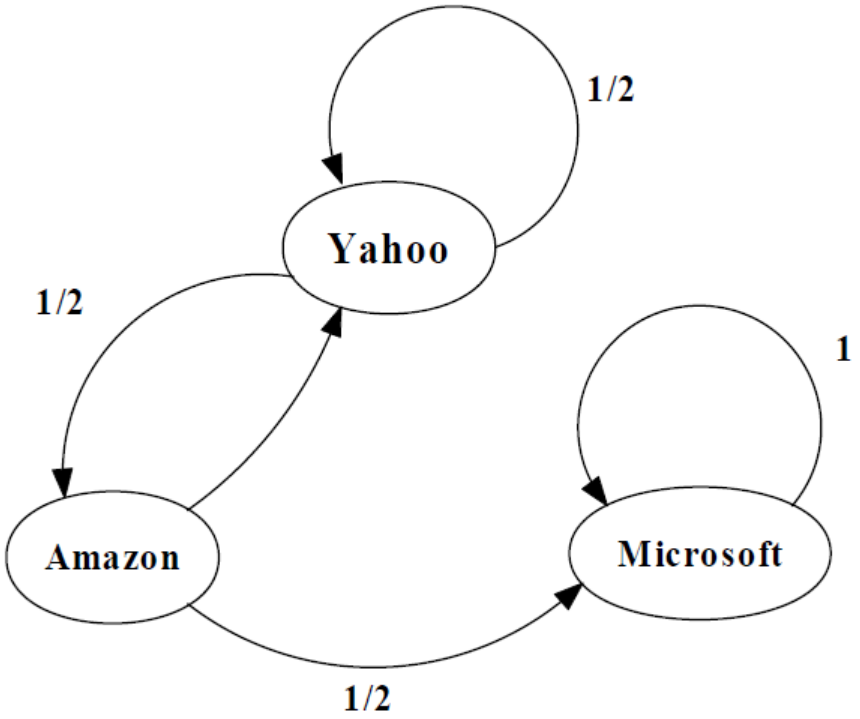
一个例子



$$\begin{bmatrix} \text{yahoo} \\ \text{Amazon} \\ \text{Microsoft} \end{bmatrix} = \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix}$$

$$\begin{bmatrix} 1/3 \\ 1/6 \\ 1/2 \end{bmatrix} = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 0 \\ 0 & 1/2 & 1 \end{bmatrix} \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix}$$

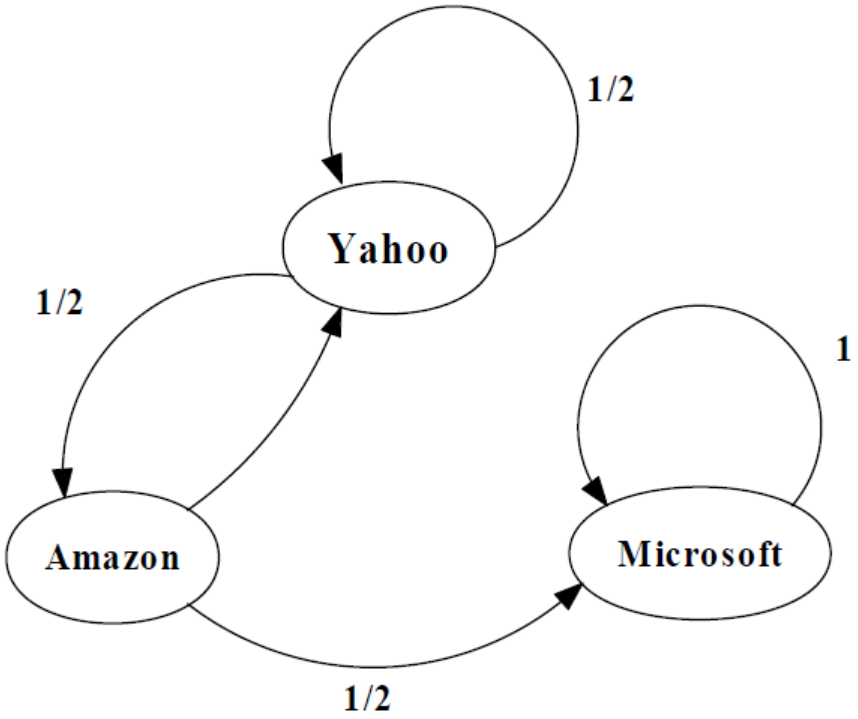
一个例子



$$\begin{bmatrix} \text{yahoo} \\ \text{Amazon} \\ \text{Microsoft} \end{bmatrix} = \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix}$$

$$\begin{bmatrix} 1/4 \\ 1/6 \\ 7/12 \end{bmatrix} = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 0 \\ 0 & 1/2 & 1 \end{bmatrix} \begin{bmatrix} 1/3 \\ 1/6 \\ 1/2 \end{bmatrix}$$

一个例子



$$\begin{bmatrix} \text{yahoo} \\ \text{Amazon} \\ \text{Microsoft} \end{bmatrix} = \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix}$$

$$\begin{bmatrix} 5/24 \\ 1/8 \\ 2/3 \end{bmatrix} \begin{bmatrix} 1/6 \\ 5/48 \\ 35/48 \end{bmatrix} \quad .... \quad \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

改进的PageRank公式

随机冲浪或随机游走(Random Walk)模型：到达u的概率由两部分组成：一部分是直接随机选中的概率(1-*d*)或(1-*d*)/*N*，另一部分是从指向它的网页顺着链接浏览的概率，则

$$R(u)=(1-d)+d\sum_{v\in B_u}\frac{R(v)}{N_v} \quad \text{或} \quad R(u)=\frac{(1-d)}{N}+d\sum_{v\in B_u}\frac{R(v)}{N_v}$$

上述两个公式中，后一个公式所有网页PageRank的和为1，前一个公式的PageRank和为$N(1-d)+d$ 。

可以证明，PageRank是收敛的。计算时，PageRank很难通过解析方式求解，通常通过迭代方式求解。*d*通常取0.85

PageRank面对的Spamming问题

- SEO（Search Engine Optimization)：通过正当或者作弊等手段提高网站的检索排名(包括PageRank)排名。

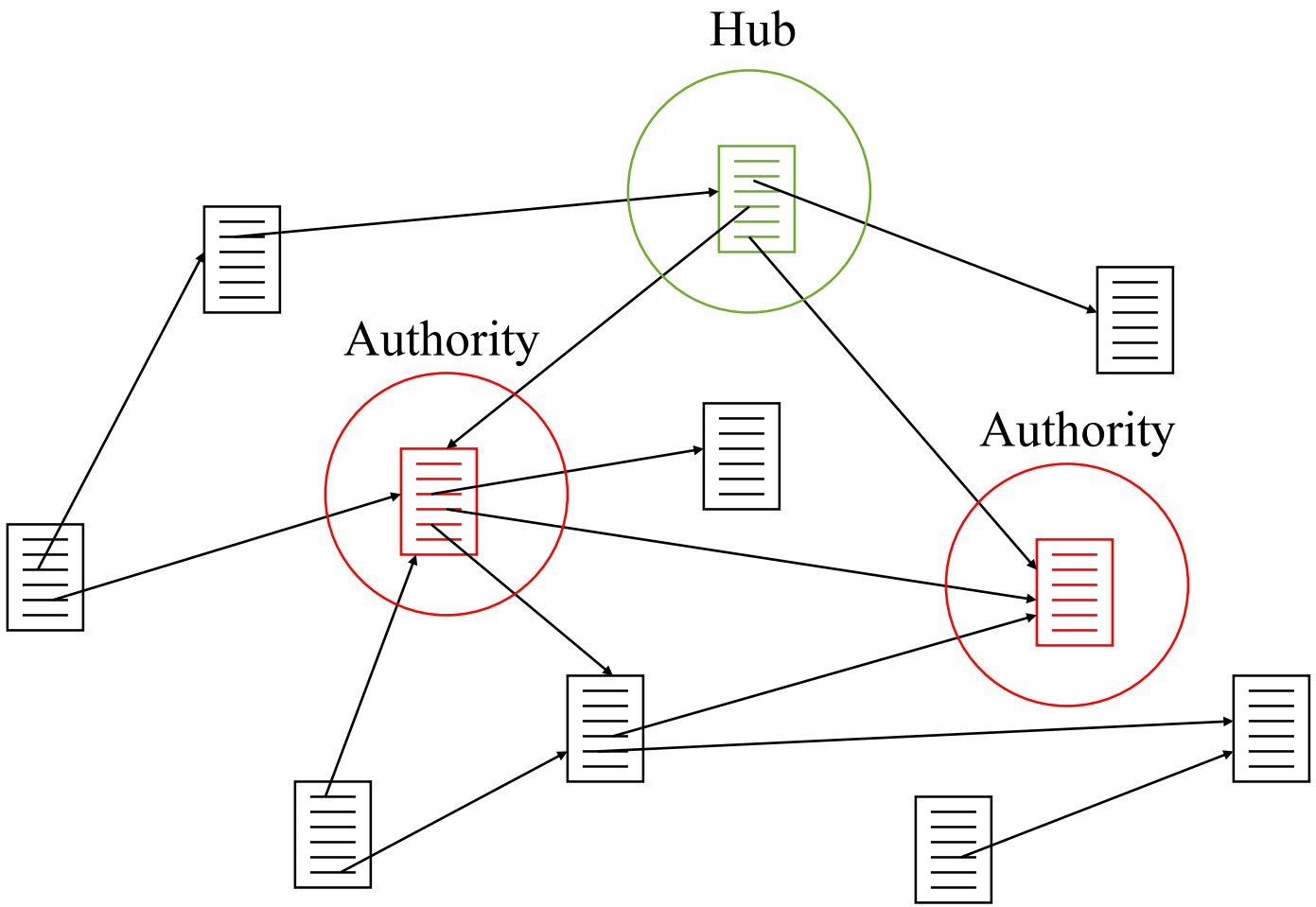
- 因此，实际中的PageRank实现必须应对这种作弊，实际实现复杂得多。实际中往往有多个因子(比如内容相似度)的融合。

IBM的HITS算法

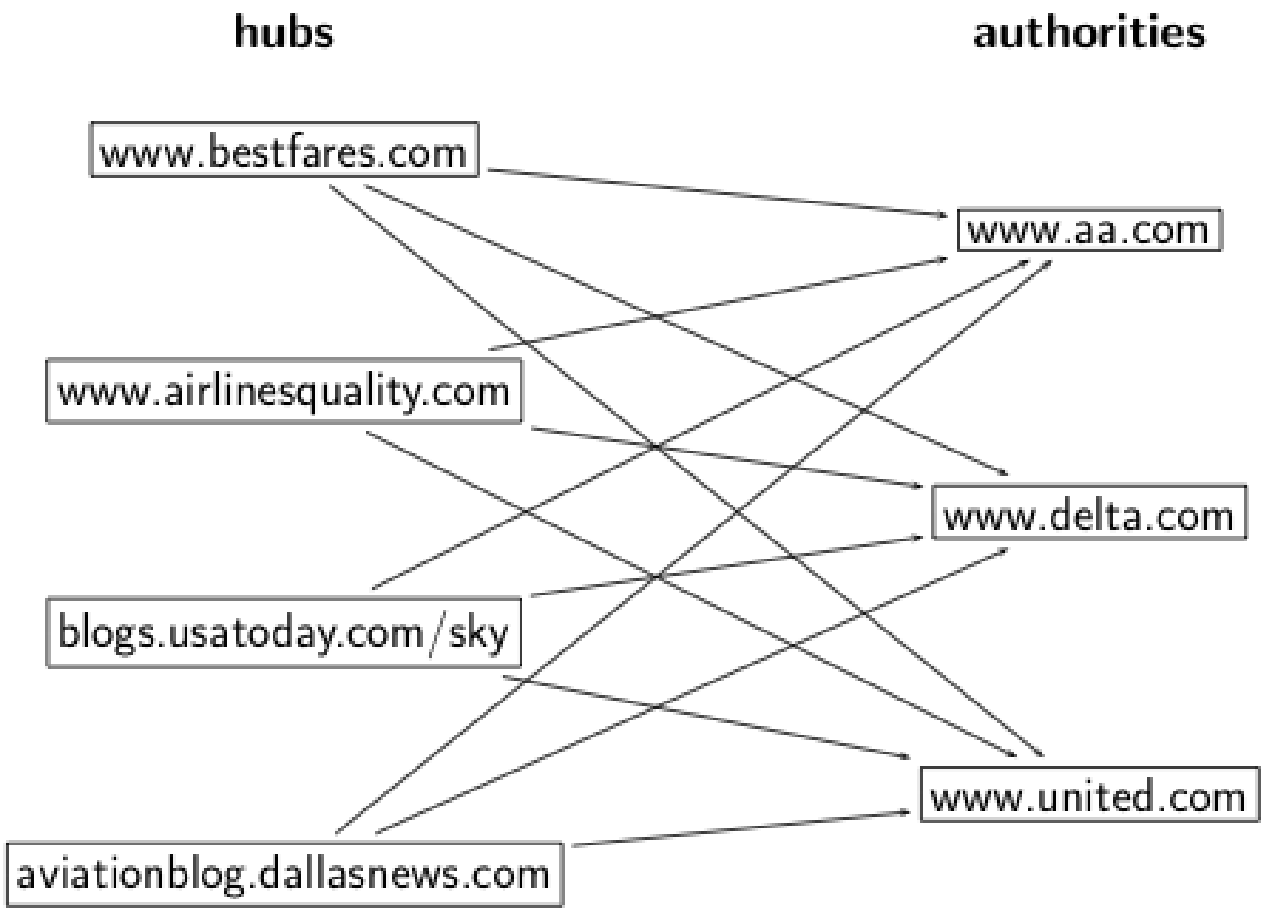
- HITS(Hyperlink-Induced Topic Search)

- 每个网页计算两个值
  - Hub：作为目录型或导航型网页的权重
  - Authority：作为权威型网页的权重

Hub & Authority示意



例子


hubs
authorities
www.bestfares.com
www.airlinesquality.com
blogs.usatoday.com/sky
aviationblog.dallasnews.com
www.aa.com
www.delta.com
www.united.com

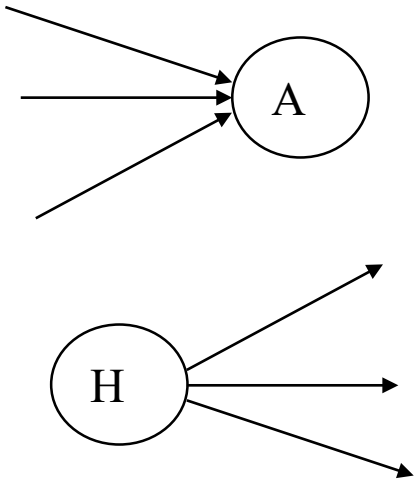
计算方法

$$A(p)=\sum H(q_i)$$

(其中q_i是所有链接到p的页面)

$$H(p)=\sum A(r_i)$$

(其中r_i是所有页面p链接到的页面)



(1) 一个网页被越重要的导航型网页指向越多，那么它的Authority越大；

(2) 一个网页指向的高重要度权威型网页越多，那么它的Hub越大。

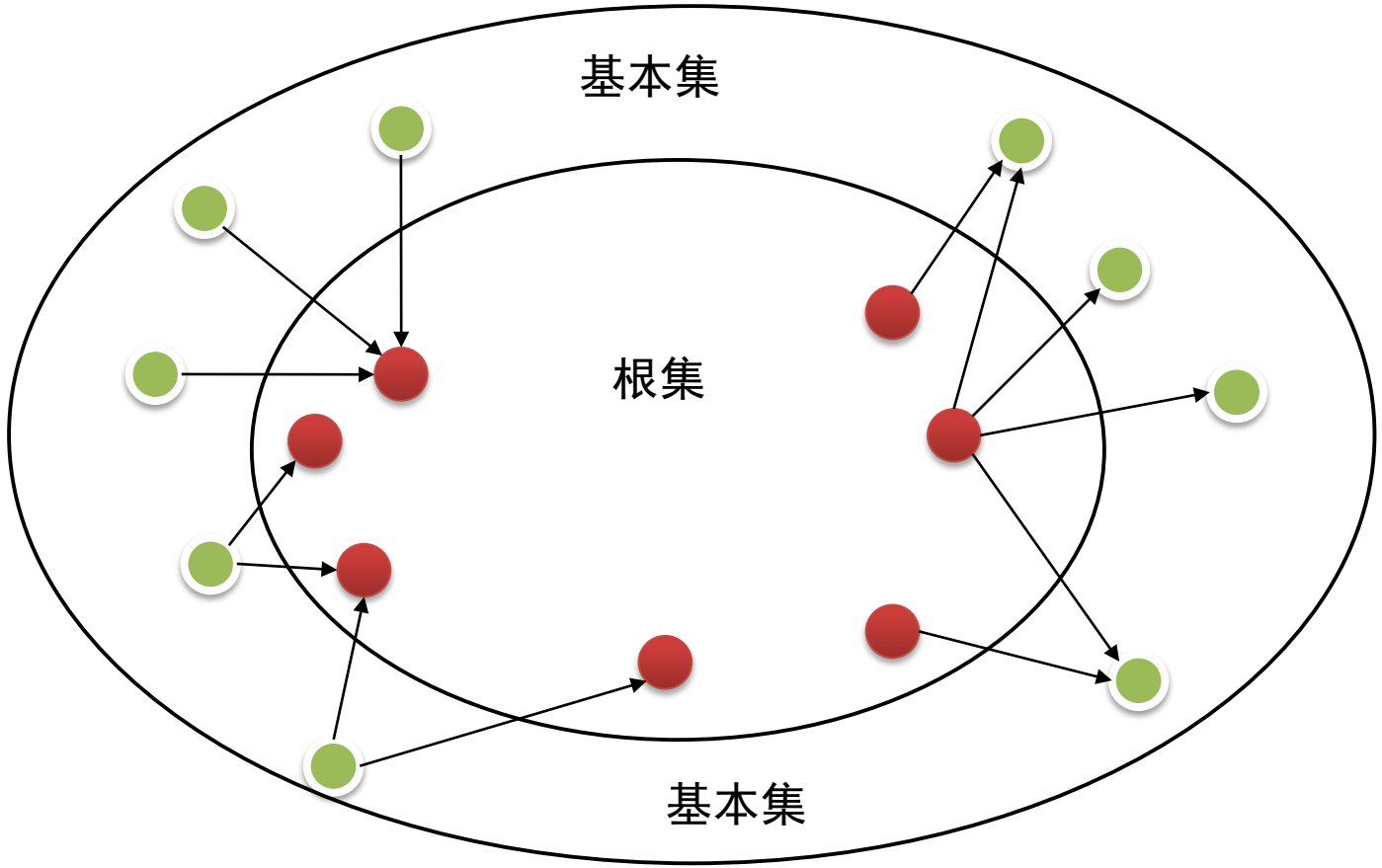
HITS算法也是收敛的，也可以通过迭代的方式计算。

HITS的计算过程

- 首先进行Web搜索；
- 搜索搜索的结果称为根集(root set)；
- 将所有链向种子集合和种子集合链出的网页加入到种子集合；
- 新的更大的集合称为基本集(base set)；
- 最后，在基本集上计算每个网页的hub值和authority值 (该基本集可以看成一个小的Web图)。

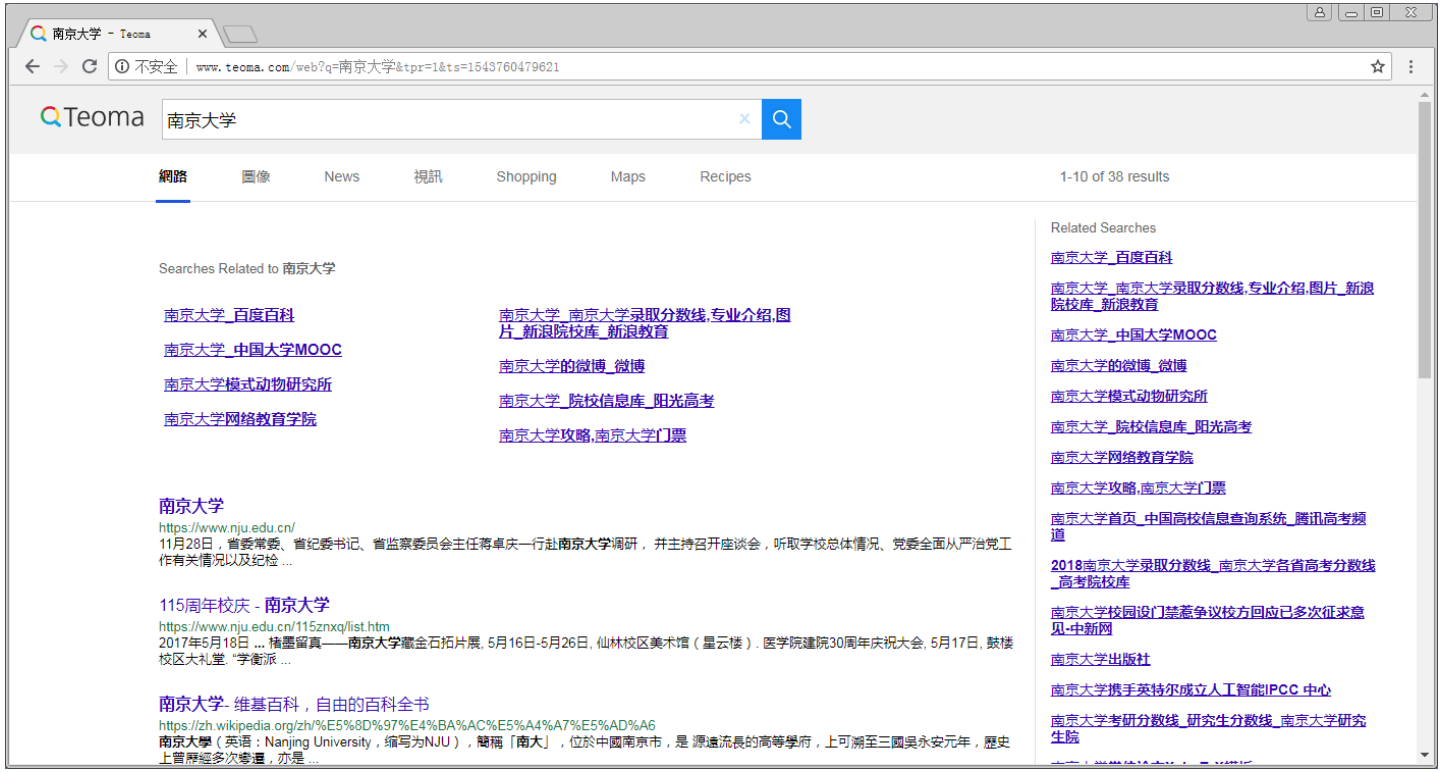
根集和基本集

- 根集往往包含200-1000个节点
- 基本集可以达到5000个节点



HITS缺点

- 计算效率低
  - 实时计算
- 主题漂移
  - 扩展集可能与主题无关
- 易作弊
- 结构不稳定
  - 删改个别少数关系，结果可能变化大



PageRank vs. HITS

- 网页的PageRank与查询主题无关，可以事先算好，因此适合于大型搜索引擎的应用。
- HITS算法的计算与查询主题相关，检索之后再进行计算，因此，不适合于大型搜索引擎。

后续

- Web文本结构分析
- 正文提取
- 复杂网络分析
- ……

小结