复旦大學
FUDAN UNIVERSITY

文本存储与检索

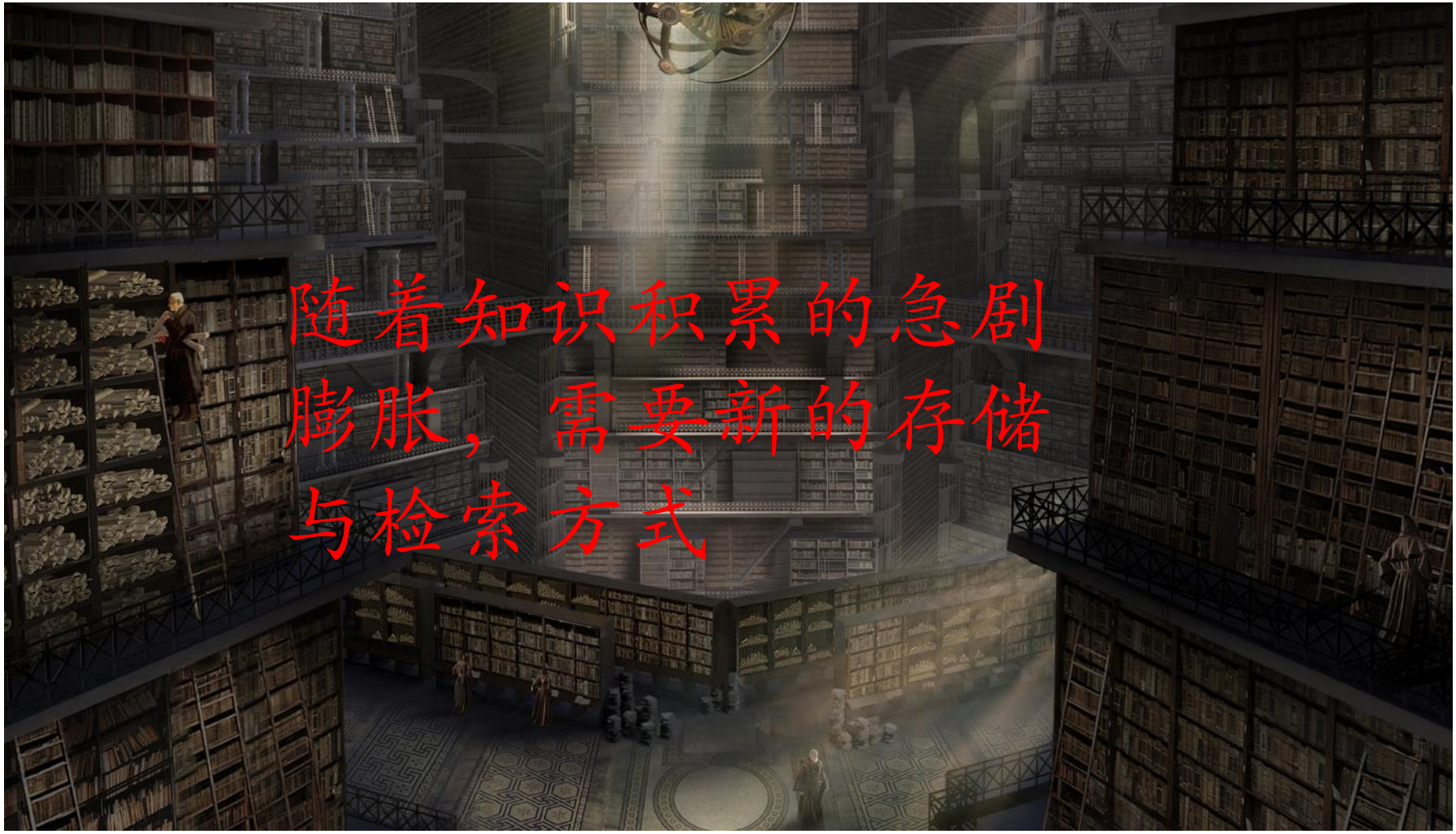
邱锡鹏
复旦大学
http://nlp.fudan.edu.cn/xpqiu

重要性

- 人类需要不断地获取、积累、交换知识
- 知识主要以文本形式存在
  - 最早可以追溯到古代的书籍编目

- 两个关键问题
  - 存储
  - 检索

早期的文本存储与检索



计算机出现以后

- 1948年C. N. Mooers在其MIT硕士论文中第一次使用了“Information Retrieval”这个术语。
- 1960－70年代在建立文摘检索系统中
  - 布尔模型(Boolean Model)
  - 向量空间模型(Vector Space Model)
  - 概率检索模型(Probabilistic Model)
- 1980年代出现商用数据库检索系统

互联网出现以后

- 1986年Internet正式形成
- 1990s第一个网络搜索工具：1990年加拿大蒙特利尔大学开发的FTP搜索工具Archie。
- WEB搜索引擎
  - 1994年美国CMU开发的Lycos。
  - 1995斯坦福大学博士生创立Yahoo。
  - 1998斯坦福大学博士生创立的Google
    - PageRank
  - 2001年李彦宏创立百度
    - 竞价排名

现代文本存储与检索

基本检索 | 多字段检索 | 多库检索 | 高级检索 | 通用命令语言检索 | 分类浏览 | 标签浏览

高级检索

检索字段	键入检索词或词组	词邻近?	命中记录数
所有字段 ▼		否 是	
所有字段 ▼		否 是	
所有字段 ▼		否 是	
数据库	中文文献库 ▼		
点击命中记录总数查看记录.		总数:	

确 定 清 除

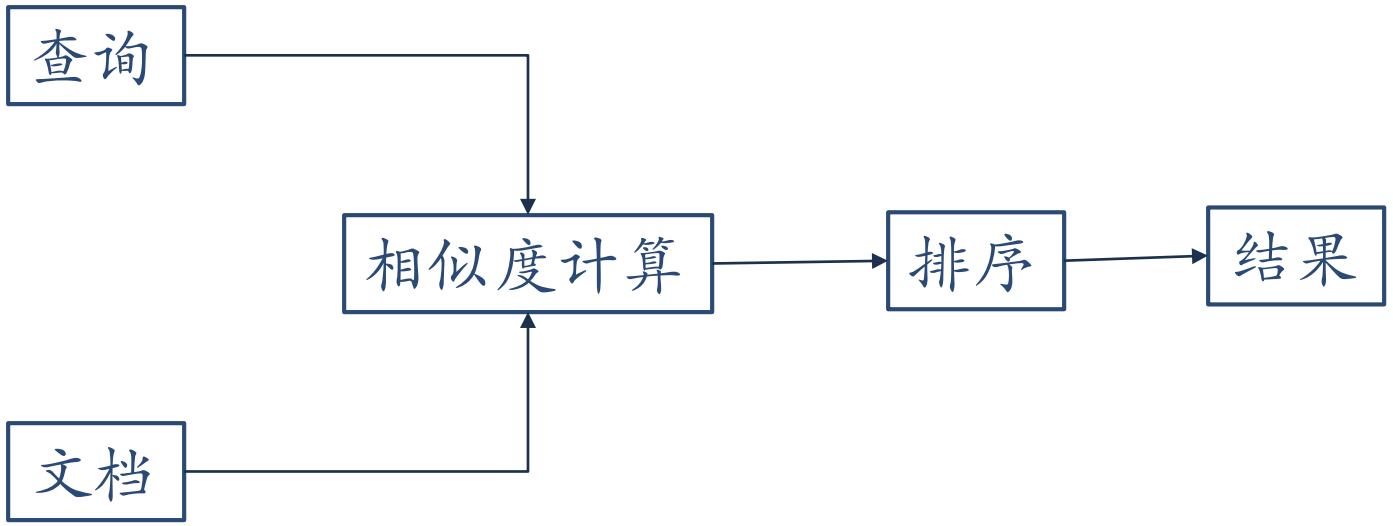
检索限制: 语言: 全部 ▼ 开始年份: 结束年份: yyyy（当不使用起/止时，使用 ？作截词）

资料类型: 全部 ▼ 分馆 ▼

信息检索

- 信息检索
  - 将信息按一定的方式组织和存储起来
  - 根据用户的需要查找相关信息
  - 包括
    - 结构化信息检索
      - 数据库检索
    - 非结构化信息检索
      - 图像检索
      - 视频检索
      - 音乐检索
      - 文本检索

文本检索 系统结构



核心问题

- 匹配问题
  - 文本相似度计算（参考 上一章内容）
- 效率问题
  - 文档索引

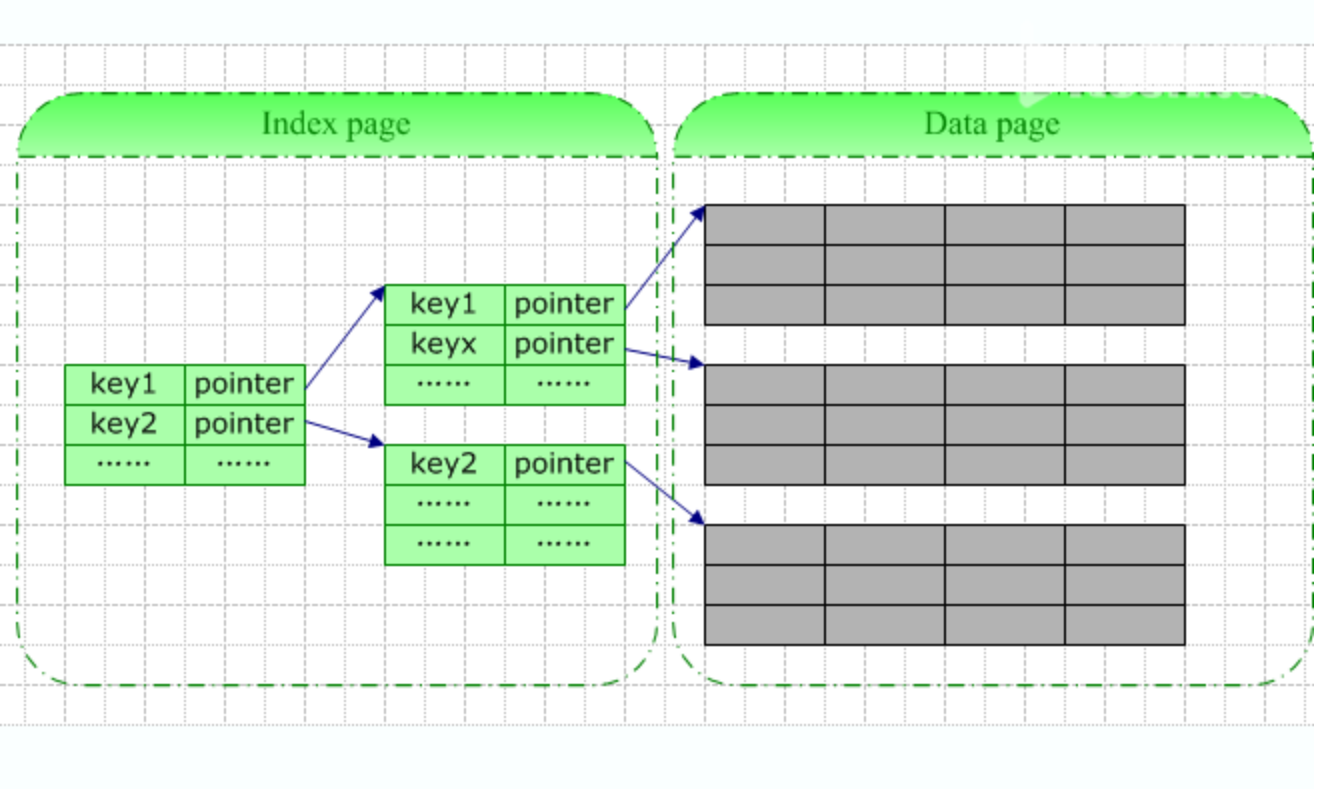
组织与存储

索引（indexing）


5F
楼层索引
Floor Index
13F 耳鼻咽喉科病房、眼科病房 Otolaryngology department ward Eye ward
12F 胃肠外科病房、痔瘘肛肠外科病房 Gastrointestinal surgery ward ZhiLou anorectal surgical ward
11F 肝胆胰脾外科病房、渠县腹腔镜治疗中心 Hepatobiliary surgical ward Quxian laparoscopic treatment center
10F 泌尿外科病房、小儿外科病房、渠县泌尿外科微创治疗中心
9F 骨科病房（关节脊柱、四肢伤病）、骨科康复治疗病房 Orthopedics ward(Joints of the spine Limb injury),Orthopaedic rehabilitation wards
8F 皮肤、医学美容科、儿童皮肤科、手术室设备层
7F 手术室、麻醉复苏室 Operating Room,Anesthesia recovery room
6F 外科重症监护病房（外科ICU）、麻醉科 Surgery Intensive Care Unit,department of anesthesiology
5F 神经外科病房、心胸外科病房 neurosurgical unit,The cardiothoracic surgery ward
4F 妇科病房、产科病房、渠县妇科腹腔镜治疗中心 Gynecology,Obstetrics,Quxian County gynecological laparoscopic treatment center
3F 新生儿监护病房（NICU）、妇产科产房、大外科办公室 Neonatal intensive care unit,Obstetrics and gynecology department,Surgical office
2F 儿科病房、儿童重症监护病房（PICU） The pediatric ward,Pediatric intensive care unit
1F
1100 mm
150 mm
900 mm
600 mm
600.00 mm
楼层索引
Floor Index
13F 耳鼻咽喉科病房、眼科病房 Otolaryngology department ward Eye ward
12F 胃肠外科病房、痔瘘肛肠外科病房 Gastrointestinal surgery ward ZhiLou anorectal surgical ward
11F 肝胆胰脾外科病房、渠县腹腔镜治疗中心 Hepatobiliary surgical ward Quxian laparoscopic treatment center
10F 泌尿外科病房、小儿外科病房、渠县泌尿外科微创治疗中心
9F 骨科病房（关节脊柱、四肢伤病）、骨科康复治疗病房 Orthopedics ward(Joints of the spine Limb injury),Orthopaedic rehabilitation wards
8F 皮肤、医学美容科、儿童皮肤科、手术室设备层
7F 手术室、麻醉复苏室 Operating Room,Anesthesia recovery room
6F 外科重症监护病房（外科ICU）、麻醉科 Surgery Intensive Care Unit,department of anesthesiology
5F 神经外科病房、心胸外科病房 neurosurgical unit,The cardiothoracic surgery ward
4F 妇科病房、产科病房、渠县妇科腹腔镜治疗中心 Gynecology,Obstetrics,Quxian County gynecological laparoscopic treatment center
3F 新生儿监护病房（NICU）、妇产科产房、大外科办公室 Neonatal intensive care unit,Obstetrics and gynecology department,Surgical office
2F 儿科病房、儿童重症监护病房（PICU） The pediatric ward,Pediatric intensive care unit
1F
1000.00 mm

数据库索引

- 在关系数据库中，索引是为了提高数据的检索效率而创建的一种分散的存储结构。

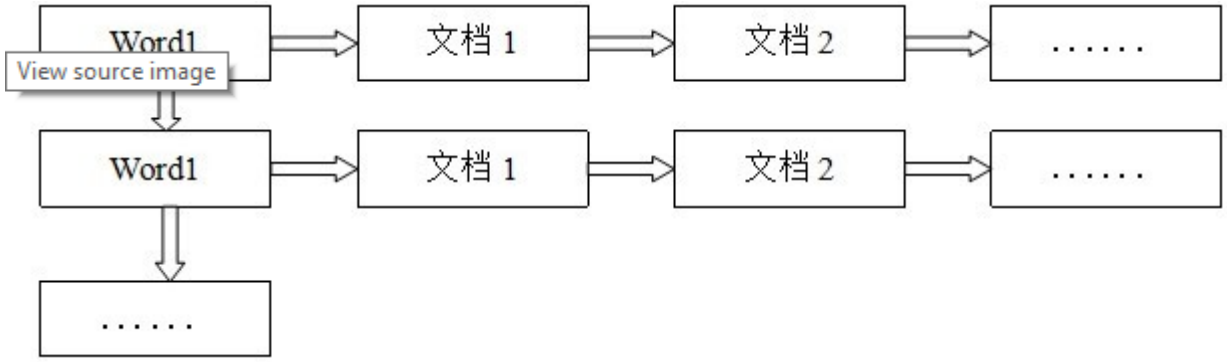


文档索引

- 索引的目的在于提高检索效率
  - 没有索引就只能对所有文档内容进行顺序匹配
- 索引对象
  - 词（Term）

倒排索引(inverted index)

- 倒排索引用来记录有哪些文档包含了某个单词。
  - 以词（Term）为核心对文档进行索引
  - 记录包含某个词的文档编号、该词在每个文档中出现的次数（TF）及出现位置等信息，这些信息称为倒排索引项。



文档索引

- 倒排索引的优势
  - 关键词个数比文档少，因此检索效率高
  - 特别适合信息检索
    - 查询词一般很少，通过几次查询就能找出所有可能的文档
- 倒排索引的数据结构
  - 关键词查询般采用B-Tree或哈希表
  - 文档列表组织一般采用二叉搜索树

文档索引

- 索引压缩
  - 减小索引大小
- 动态索引
  - 索引库的动态维护、更新
- 分布式索引
  - 索引信息分布在不同机器上

一个相关话题：网页爬虫

- 网页爬虫
  - Web Crawler, spider
  - 快速有效地收集尽可能多的有用Web页面，包括页面之间的链接结构

- 策略
  - 深度优先
  - 广度优先
  - 实际应用中以广度优先为主，深度优先为辅

检索

三个检索模型

- 布尔模型(Boolean Model)
- 向量空间模型(Vector Space Model)
- 概率检索模型(Probabilistic Model)

布尔模型

- 一种简单的检索模型，它建立在经典的集合论和布尔代数的基础上
- 系统索引词集合中的每一个索引词在一篇文档中只有两个状态
  - 出现
  - 不出现
  - 检索提问式q由三种布尔运算符“and”、“or”、“not”连接索引词来构成

布尔模型

- 相似度计算
  - 查询布尔表达式和所有文档的布尔表达式进行匹配，匹配成功的文档的得分为1，否则为0
  - 类似于传统数据库检索，是精确匹配

- 例子
  - http://202.120.227.11/F/?func=find-d-0

参考上一章内容

向量空间模型

- 将查询Q和文档D_j都表示为向量
  - $q=(w_{1,q}, w_{2,q}, \cdots, w_{t,q})$
  - $dj=(w_{1,j}, w_{2,j}, \cdots, w_{t,j})$
    - 其中，t是系统中所有索引项的数目，
- 向量模型通过d_j和q来评价文档D_j和查询Q的相关度。
  - 一般使用两个向量之间的夹角余弦值来计算

概率检索模型

- 概率检索模型试图在一个概率框架中处理信息检索问题。
  - 其基本思想是：给定一个查询请求q和集合中的文档d_j，估计查询请求与文档d_j相关的概率，
    - 只依赖于查询请求和文档。
  - 最早由Maron和Kuhn在1960年提出。
  - 是目前效果最好的模型之一

概率检索模型

- 定义
  - R表示已知的相关文档集（或最初的猜测集），用
  - $\bar{R}$表示R的补集。
  - $P(R|d_j)$表示文档d_j与查询q相关的概率
  - $P(\bar{R}|d_j)$表示文档d_j与查询q不相关的概率。
- 文档dj与查询q的相似度$sim(d_j, q)$为：

$$sim(d_j, q) = \frac{P(R \mid d_j)}{P(\overline{R} \mid d_j)}$$

概率检索模型

- 根据贝叶斯定理有

$$sim(d_j, q) = \frac{P(d_j \mid R) \times P(R)}{P(d_j \mid \overline{R}) \times P(\overline{R})}$$

概率检索模型

- 假设标引词独立，则

$$sim(d_j, q) \sim \frac{(\prod_{g_i(d_j)=1} P(k_i \mid R)) \times (\prod_{g_i(d_j)=0} P(\bar{k}_i \mid R))}{\prod_{g_i(d_j)=1} P(k_i \mid \bar{R})) \times (\prod_{g_i(d_j)=0} P(\bar{k}_i \mid \bar{R})}$$

- 这是概率模型中排序计算的主要表达式

概率检索模型

- 取对数，在相同背景下，忽略对所有因子保持恒定不变的因子，则有

$$sim(d_j, q) \sim \sum_{i=1}^{t} w_{iq} \times w_{ij} \times \left(\log \frac{P(k_i \mid R)}{1 - P(k_i \mid R)} + \log \frac{1 - P(k_i \mid \bar{R})}{P(k_i \mid \bar{R})} \right)$$

概率检索模型

- 令V是初始检索结果的子集，有r个，取自检索结果集中前r个文档，这些检索结果是经过概率模型排好顺序的
- 令Vi是V中所有包含索引项ki的那些文档，显然Vi是V的子集；为简单起见，直接用V和Vi表示这些集合中的元素数量
- 修改对概率 $P(k_i \mid R)$和 $P(\bar{k}_i \mid R)$的计算方法

$$P(k_i \mid R) = \frac{V_i}{V} \qquad P(k_i \mid \bar{R}) = \frac{n_i - V_i}{N - V}$$

概率检索模型

- 为保证数值计算的稳定性，常用下列公式计算相似度：

$$P(k_i \mid R) = \frac{V_i + 0.5}{V + 1} \qquad P(k_i \mid \overline{R}) = \frac{n_i - V_i + 0.5}{N - V + 1}$$

$$P(k_i \mid R) = \frac{V_i + \frac{n_i}{N}}{V + 1} \qquad P(k_i \mid \overline{R}) = \frac{n_i - V_i + \frac{n_i}{N}}{N - V + 1}$$

BM25

- 对于查询Q，包含词 $q_1,\ldots,q_n$，文档D 的 BM25 得分为

$$\text{score}(D,Q) = \sum_{i=1}^{n} \text{IDF}(q_i) \cdot \frac{f(q_i, D) \cdot (k_1 + 1)}{f(q_i, D) + k_1 \cdot \left(1 - b + b \cdot \frac{|D|}{\text{avgdl}}\right)}$$

优缺点

- 优点
  - 理论上讲，文档按照其与目标集合的相关概率降序排列
- 缺点
  - 需要最初将文档分为相关和不相关的集合
  - 所有权重都是二值的，模型中仍然假设索引项之间是相互独立的
  - 相关性R，比较难以理解

改进方法：
查询条件概率模型

语言模型

- 给句子赋予概率，表示句子的可能性/合理性
  - ！在报那猫告做只
  - 那只猫在作报告！
  - 那个人在作报告！

$$P(x_1, x_2, \cdots, x_n)$$
$$= \prod_i P(x_i | x_{i-1}, \cdots, x_1)$$
$$\approx \prod_i P(x_i | x_{i-1}, \cdots, x_{i-n+1})$$

N元语言模型

查询条件概率模型

- 基于语言模型的检索

- 方法
  - 对每个文档都估计一个语言模型
  - 估计$P(q|M_{d_i})$,根据文档di的语言模型生成查询q的概率
  - 根据$P(q|M_{d_i})$对文档进行排序

总结

- 布尔模型(Boolean Model)
- 向量空间模型(Vector Space Model)

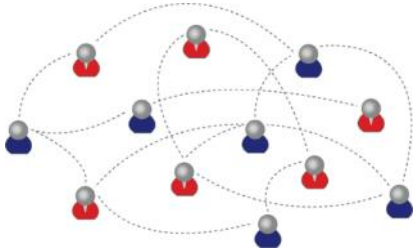
- 概率检索模型(Probabilistic Model)
  - BM25
  - 基于统计语言模型的检索模型

排序

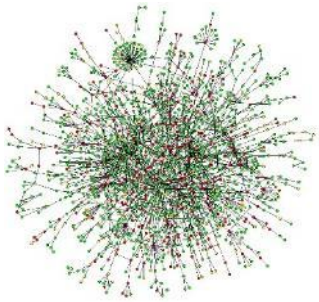
问题

- 如何让一个网页在搜索结果中排名更靠前?

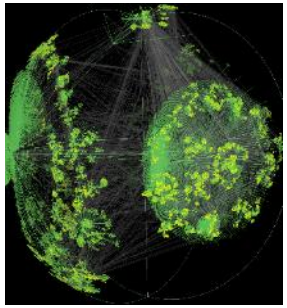
图表示



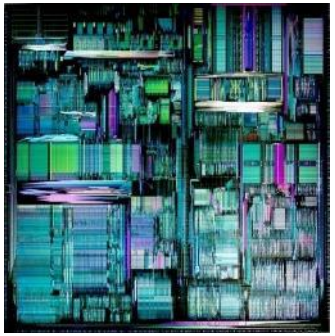
Social networks



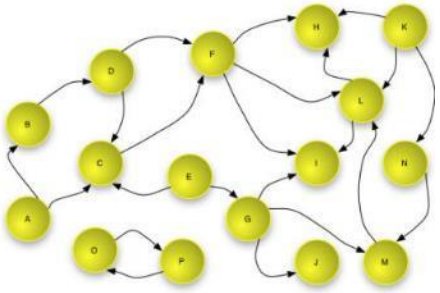
Protein Interactions



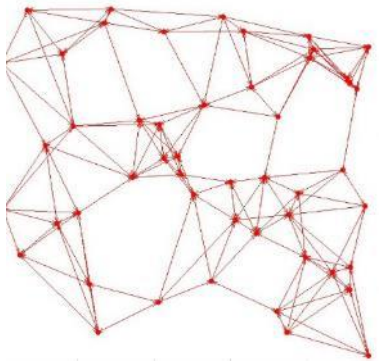
Internet



VLSI networks



Data dependencies



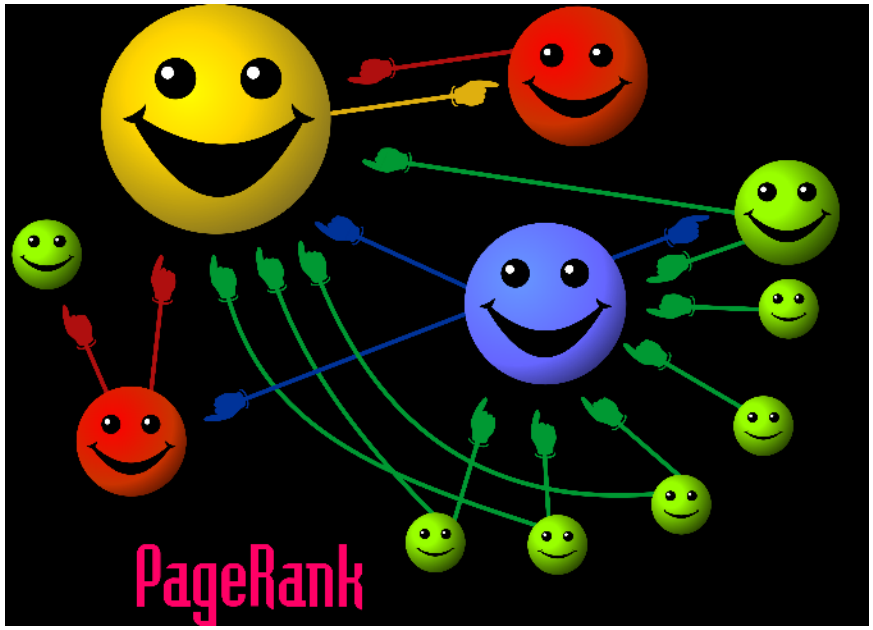
Neighborhood graphs

定义

- G = (V, E)
  - V节点
    - 度
  - E 边
    - 可以表示为连接矩阵 adjacency matrix
  - 有向、无向
- 计算
  - 两个节点之间的最短路径

PageRank™

- Google搜索引擎的核心：PageRank™算法
  - 由Larry Page和Sergey Brin提出
  - 把从A页面到B页面的链接解释为A页面给B页面投票



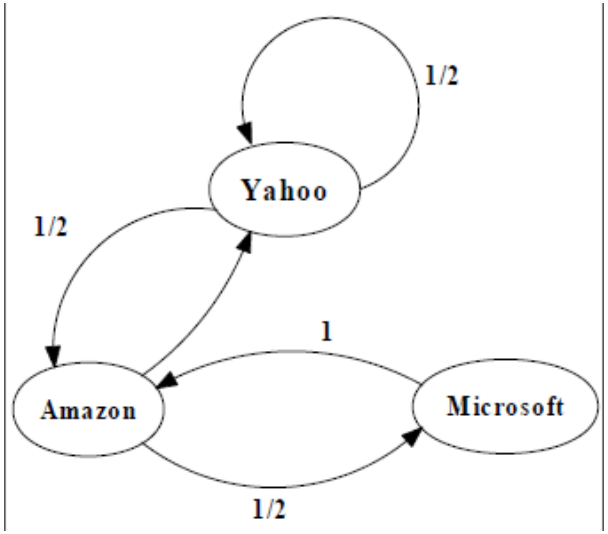
投票方法

- 迭代计算

$$PR(p_i) = \sum_{p_j \in M(p_i)} \frac{PR(p_j)}{L(p_j)}$$

- $PR(p_i)$ 网页p_i的PageRank值
- $M(p_i)$ 指向网页 p_i的网页集合
- $L(p_j)$ 网页p_j的外链数量

例子



$$M = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 1 \\ 0 & 1/2 & 0 \end{bmatrix} \cdot \begin{bmatrix} \text{yahoo} \\ \text{Amazon} \\ \text{Microsoft} \end{bmatrix} = \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix}$$

第1次迭代

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

第2次迭代

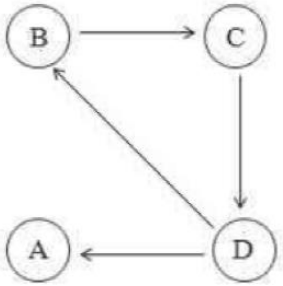
$$\begin{bmatrix} 5/12 \\ 1/3 \\ 1/4 \end{bmatrix} = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 1 \\ 0 & 1/2 & 0 \end{bmatrix} \begin{bmatrix} 1/3 \\ 1/2 \\ 1/6 \end{bmatrix}$$

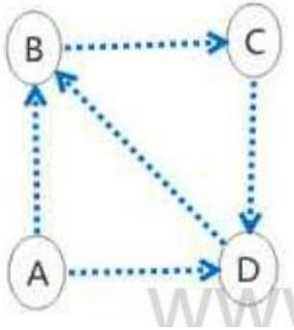
$$\begin{bmatrix} 3/8 \\ 11/24 \\ 1/6 \end{bmatrix} \begin{bmatrix} 5/12 \\ 17/48 \\ 11/48 \end{bmatrix} \ldots \begin{bmatrix} 2/5 \\ 2/5 \\ 1/5 \end{bmatrix}$$

23次迭代后收敛

存在问题

- 没有向外链接的页面
- 这些页面就像“黑洞”会吞噬掉用户继续向下浏览的概率，Rank Sink





- 增加阻尼系数（damping factor）
- PR值归一化

PageRank算法

$$PR(p_i) = \frac{1-d}{N} + d \sum_{p_j \in M(p_i)} \frac{PR(p_j)}{L(p_j)}$$

- Where:
  - d是阻尼系数，[0,1]
    - 在任意时刻，用户到达某页面后并继续向后浏览的概率，该数值是根据上网者使用浏览器书签的平均频率估算而得
  - N是所有页面的数量

PageRank算法

- PageRank值表示为向量R

$$\mathrm{R} = \begin{bmatrix} PR(p_1) \\ \mathrm{PR}(\mathrm{p}_2) \\ \vdots \\ \mathrm{PR}(\mathrm{p}_\mathrm{N}) \end{bmatrix}$$

- PageRank算法可以写为

$$\mathrm{R} = \begin{bmatrix} (1-d)/N \\ (1-d)/N \\ \vdots \\ (1-d)/N \end{bmatrix} + d \begin{bmatrix} l(p_1,p_1) & \cdots & l(p_1,p_N) \\ \vdots & \ddots & \vdots \\ l(p_N,p_1) & \cdots & l(p_N,p_N) \end{bmatrix} R$$

PageRank算法

- 矩阵形式

$$\mathrm{R} = \frac{1-d}{N} 1_{N\times 1} + dAR$$

$$\mathrm{R} = \left(\frac{1-d}{N} 1_{N\times N} + dA\right) R = UR$$

- R为矩阵U的特征值1的特征向量

HITS算法

- Jon Kleinberg 于1997 年提出
- HITS: Hyperlink-Induced Topic Search

- HITS算法最基本的两个定义
  - Hub页面（枢纽页面）
    - 包含了很多指向高质量“Authority”页面链接的网页
    - 比如：各种导航网页
  - Authority页面（权威页面）
    - 指与某个领域或者某个话题相关的高质量网页

HITS算法

- 算法基本思想：相互增强关系
- HITS 算法的两个基本假设
  - 一个好的“Authority”页面会被很多好的“Hub”页面指向；
  - 一个好的“Hub”页面会指向很多好的“Authority”页面；

HITS算法

- auth[i] == sum(hub [j] * e[j][i], j = 1..n)
- hub [i] == sum(auth[j] * e[i][j], j = 1..n)

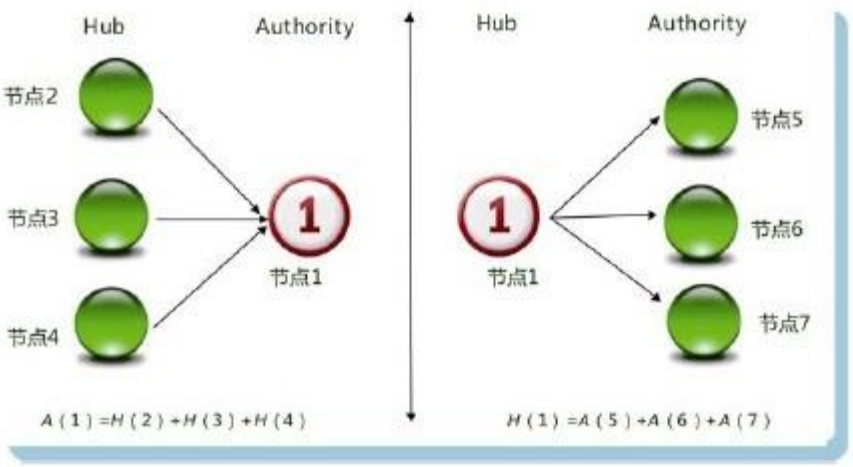


图 6-14 Hub 与 Authority 权值计算

https://en.wikipedia.org/wiki/HITS_algorithm

PageRank与HITS比较

- HITS算法是与用户输入的查询请求密切相关的，而PageRank与查询请求无关。
  - HITS算效率较低
- HITS算法的计算对象数量较少，只需计算扩展集合内网页之间的链接关系；而PageRank是全局性算法，对所有互联网页面节点进行处理；
- 从链接反作弊的角度来说，PageRank从机制上优于HITS算法，而HITS算法更易遭受链接作弊的影响。

排序算法的其他应用

- Social network (Facebook, Twitter, etc)
  - Node: Person; Edge: Follower / Followee / Friend
  - Higher PR value: Celebrity
- Citation network
  - Node: Paper; Edge: Citation
  - Higher PR values: Important Papers.
- Protein-protein interaction network
  - Node: Protein; Edge: Two proteins bind together
  - Higher PR values: Essential proteins.

思考

- 如何提高一个网页、网站的PR值?

- 一个附属产业
  - 搜索引擎优化（SEO）

其他方法

- TrustRank

- 基于机器学习的方法
  - Learn to Rank
    - RankSVM、RankNet、ListNet
    - Learning to Rank工具包
      - RankLib
      - http://people.cs.umass.edu/~vdang/ranklib.html

评价

- 评价检索模型或搜索引擎的性能
  - 搜索质量vs. 搜索效率
  - 对搜索质量的评价

- 需要
  - 评测数据集
  - 评测指标

评测数据集

- 一般人工构建
- 构成
  - 较大规模的文档集合D
  - 查询集Q及每个查询q对应的相关文档列表RELq
    - 可通过Pooling方式构建
- 一般采用国内外权威评测数据集

评价指标

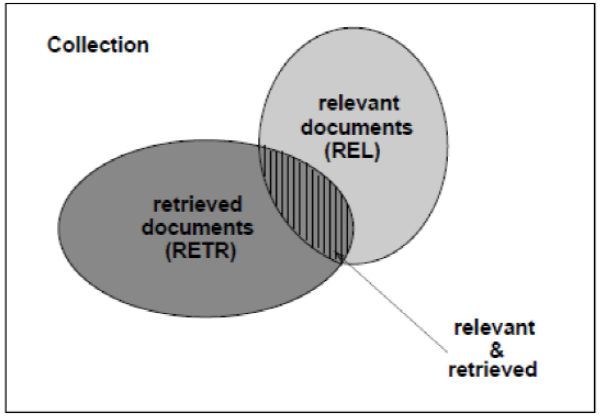
- 衡量检索结果与标准答案的一致性
  - 对非排序检索的评价
  - 对检索结果集合进行整体评价
  - 对排序检索的评价
- 考虑相关文档在检索结果中的排序位置
  - {Rel, Non-Rel, Non-Rel} 优于{Non-Rel，Non-Rel，Rel}

对非排序检索的评价

- 准确率 (precision)
  - 返回结果中相关文档数目/返回结果数目
  - 也叫精度
- 召回率 (recall)
  - 返回结果中相关文档数目/所有相关文档数目

$$precision = \frac{|\text{RETR} \cap \text{REL}|}{|\text{RETR}|}$$

$$recall = \frac{|\text{RETR} \cap \text{REL}|}{|\text{REL}|}$$




```
F1 = 2 * P * R / (P + R)
```

对排序检索的评价

- 系统检索出来的相关文档越靠前
  - p@n
  - MAP
  - NDCG
  - …

MAP(Mean Average Precision)

- 平均精度 Average Precision

$$\text{AveP} = \frac{\sum_{k=1}^{n} (P(k) \times \text{rel}(k))}{\text{number of relevant documents}}$$

  - rel(k) 是否相关
- 平均精度均值
  - 给定Q个查询

$$\text{MAP} = \frac{\sum_{q=1}^{Q} \text{AveP(q)}}{Q}$$

NDCG

https://en.wikipedia.org/wiki/Discounted_cumulative_gain

- Cumulative Gain

$$CG_p = \sum_{i=1}^{p} rel_i$$

- Discounted Cumulative Gain

$$DCG_p = \sum_{i=1}^{p} \frac{2^{rel_i} - 1}{\log_2(i+1)}$$

- Normalized DCG

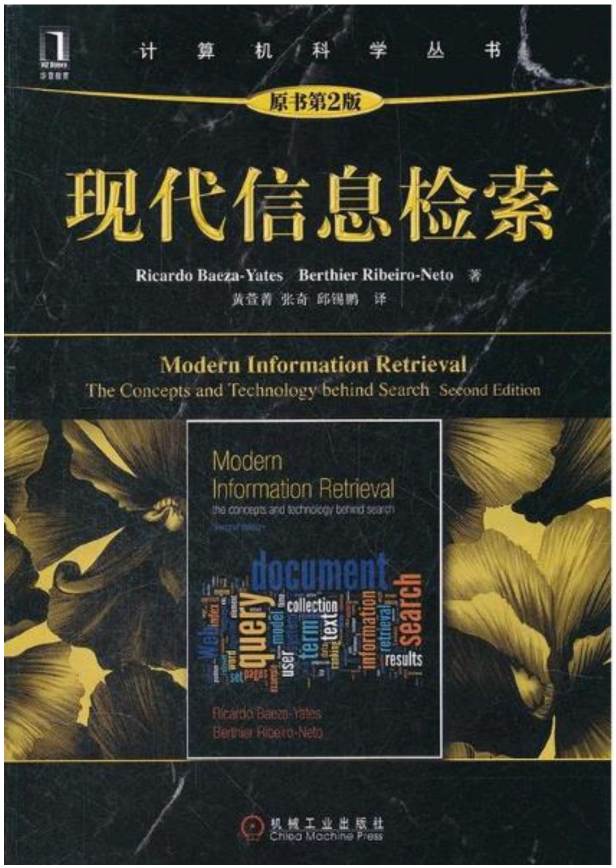
$$nDCG_p = \frac{DCG_p}{IDCG_p}$$

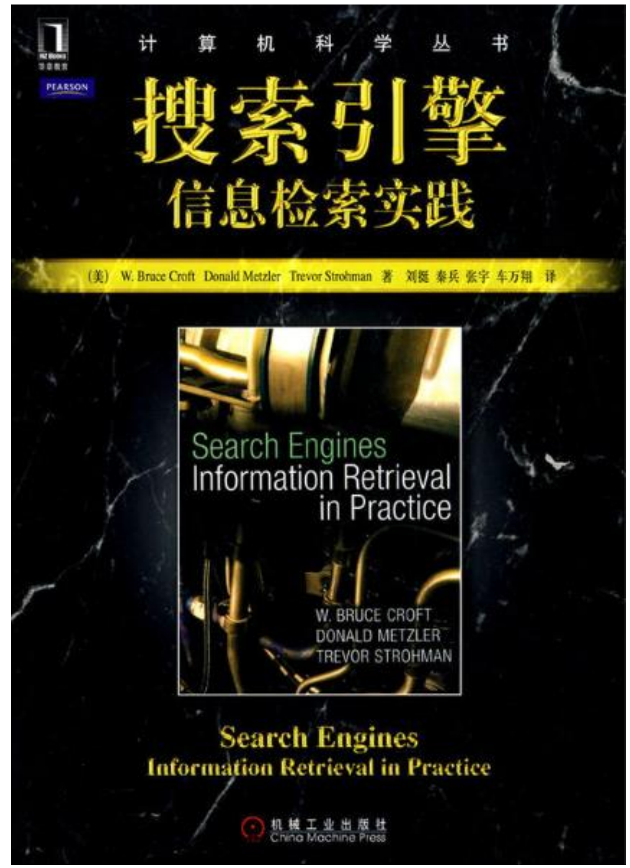
$$IDCG_p = \sum_{i=1}^{|REL|} \frac{2^{rel_i} - 1}{\log_2(i+1)}$$

总结

- 如何开发一个信息检索系统?
  - 获取
  - 存储
  - 索引
  - 检索
  - 排序
  - 评价

参考文献


计 算 机 科 学 丛 书
原书第2版
现代信息检索
Ricardo Baeza-Yates Berthier Ribeiro-Neto 著
黄萱菁 张奇 邱锡鹏 译
Modern Information Retrieval
The Concepts and Technology behind Search Second Edition
Modern
Information Retrieval
the concepts and technology behind search
document
query
collection
search
results
Ricardo Baeza-Yates
Berthier Ribeiro-Neto
机械工业出版社
China Machine Press


计 算 机 科 学 丛 书
PEARSON
搜索引擎
信息检索实践
(美) W. Bruce Croft Donald Metzler Trevor Strohman 著 刘挺 秦兵 张宇 车万翔 译
Search Engines
Information Retrieval
in Practice
W. BRUCE CROFT
DONALD METZLER
TREVOR STROHMAN
Search Engines
Information Retrieval in Practice
机械工业出版社
China Machine Press

重要的会议

- 国际会议：
  - SIGIR、ACL、WWW、SIGKDD
  - CIKM、ICML
  - TREC
  - AIRS
- 国内会议：
  - 全国信息检索及内容安全学术会议(2年一届)
  - 全国计算语言学联合会议(2年一届)

重要的期刊

- 国际
  - ACM Transactions on Information Systems (TOIS)
  - ACM Transactions on Asian Language Information Processing (TALIP)
  - Information Processing & Management (IP&M)
  - Information Retrieval
- 国内
  - 中文信息学报
  - 情报学报

重要的工具

- Lemur
  - 包含各种IR模型的实验平台，C++
- SMART
  - 向量空间模型工具, C编写
- Lucene
  - 开源检索工具，各种语言编写的版本

项目实践

- 用Lucene搭建一个中文维基百科的搜索引擎
  - Wiki语料：http://pan.baidu.com/s/1geGozC3
  - Lucene： http://lucene.apache.org/

- 参考代码
  - https://github.com/FudanNLP/fnlp/tree/master/fnlp-app

谢 谢

如果您有任何意见、评论以及建议，请通过GitHub的Issues页面进行反馈。