第三章：

文本检索技术

杨建武

北京大学计算机科学技术研究所

Email:yangjianwu@icst.pku.edu.cn

信息检索概述

信息爆炸

- 网页数量：
  - 1997：3.2亿
  - 1999：8亿(15TB)
  - 2002：20亿(Google)
  - 2004：43亿(Google)
  - 2006:100亿(Google)

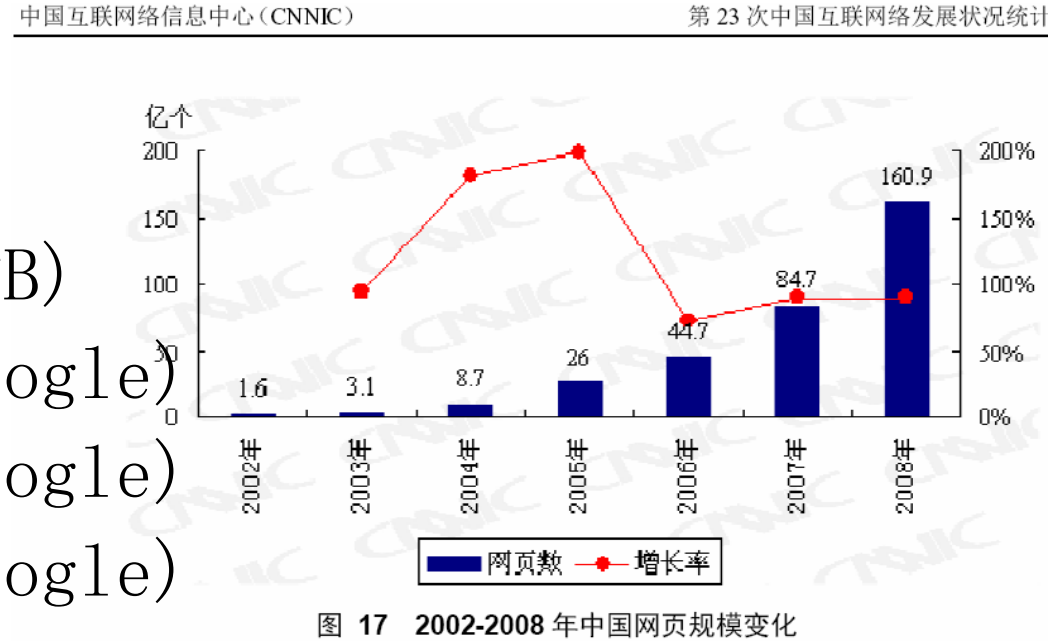
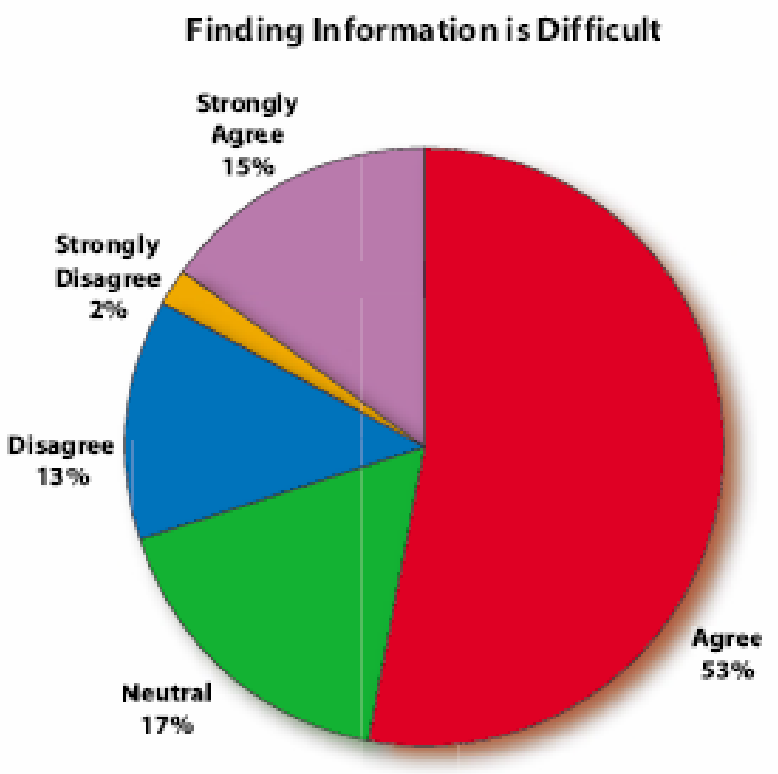


图 17　2002-2008 年中国网页规模变化

- recently study shows:
  - 1TB/year for books,
  - 2TB/year for newspapers,
  - 1TB/year for periodcals,
  - 19TB/year for office docs.

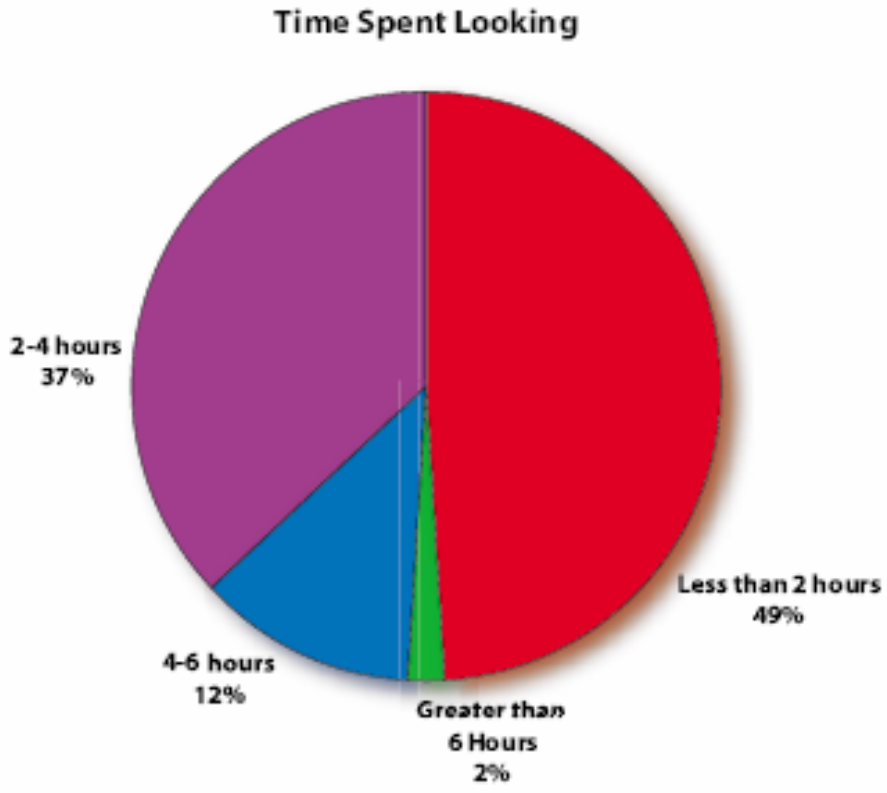
信息检索花费大量时间

Question #1 – "Finding the information I need to do my job is difficult: agree – disagree"



68%的人认为信息难找

Question #2 – "How much time do you spend each day searching for information that is critical to your job performance?"



51%的人每天至少花2个小时检索信息

信息检索的定义

- information retrieval
- 广义的信息检索是指将信息按一定的方式组织和存储起来，并根据信息用户的需要找出有关的信息的过程和技术。
  - 全称：信息存储与检索（information storage and retrieval）。
- 狭义的信息检索则仅指该过程的后半部分，即从信息集合中找出所需要的信息的过程，相当于人们所说的信息查询（information search）。

信息检索的发展

- 信息检索通常是指对无结构数据（如自然语言文本）的检索
- 图片，音频、视频文件等多媒体数据也是无结构的，不在传统IR研究范围之内。但近年来逐渐纳入IR领域内，如TREC引入了SDR (Speech Document Retrieval) Track和Video Track。
- 信息检索的发展阶段
  - 手工检索(早期,情报检索)
  - 穿孔卡片检索(1950s)
  - 计算机检索(面向主题,1960s)
  - 联机检索（1970s,1980s)
  - Web检索(1990s)

基于分类的检索

- 以分类体系为基础，将各种资料的按信息的类别进行分类和系统排列，并编排组织成一个完整的体系。
- 按照知识门类的逻辑次序，运用概念划分和归属的方法，由总到分，逐级展开，形成一个严格有序的等级制体系

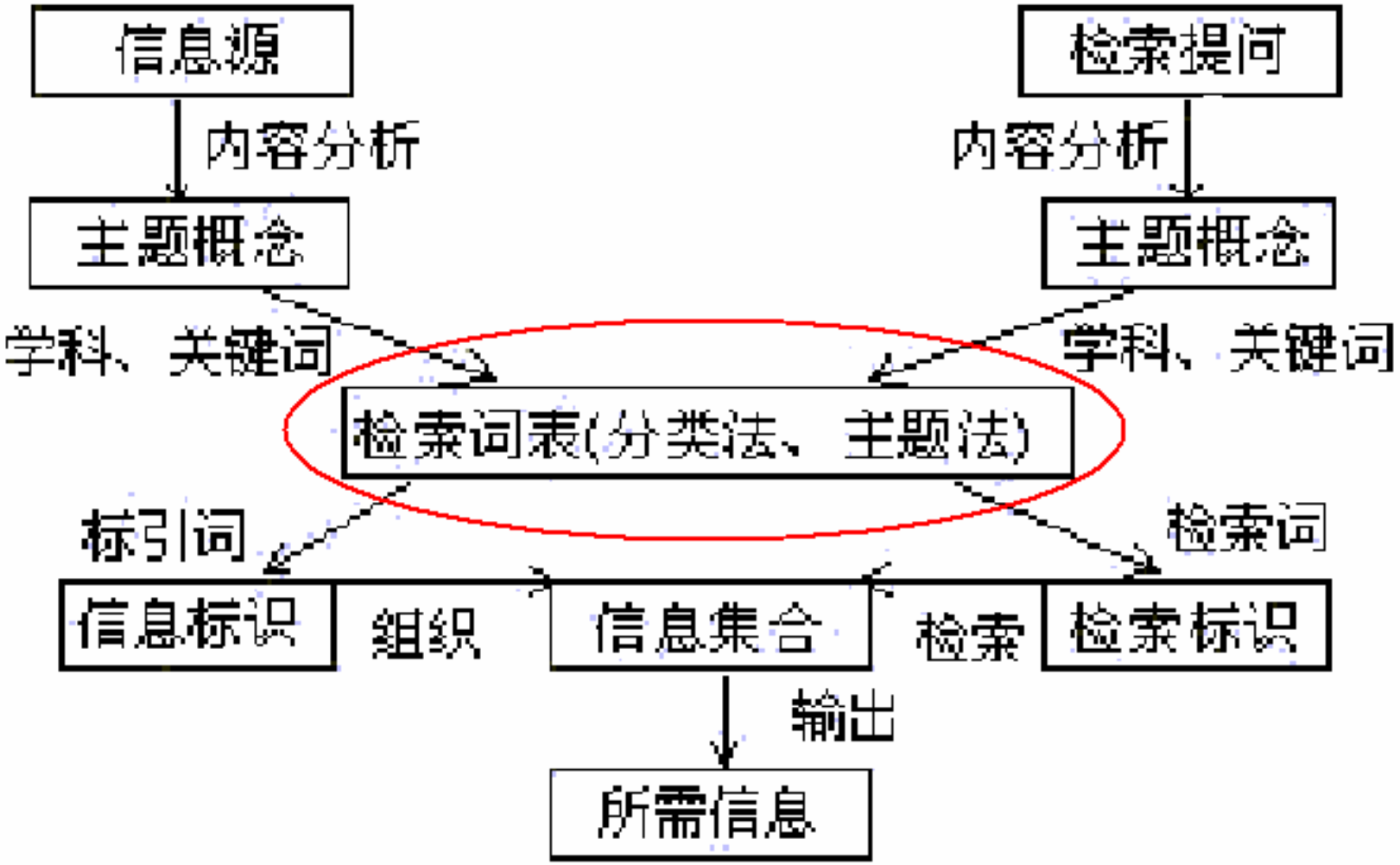
分类的例子

- A 马、列、毛思想、邓小平理论
- B 哲学、宗教
- 社会科学——C 社会科学总论
- D 政治、法律
- E 军事
- F 经济
- G 语言、文字
- I 文学
- J 艺术
- K 历史、地理
- 自然科学——N 自然科学总论

分类检索的不足

- 检索者检索时首先必须了解分类体系才能顺利查找到相应的类目，如果不熟悉分类体系，会带来一定的困难 。
- 体系分类采用列举类目的方法，受到类目数量的限制，缺乏专指性，查准率不高。
- 由于分类表的结构是固定的，不便于随时修订和增设新的类目。
- 体系分类语言采用分类号作为检索标识，检索文献时，需要将检索文献的主题内容转换成分类号，转换过程中，容易产生误差，造成误检。

基于主题词的检索

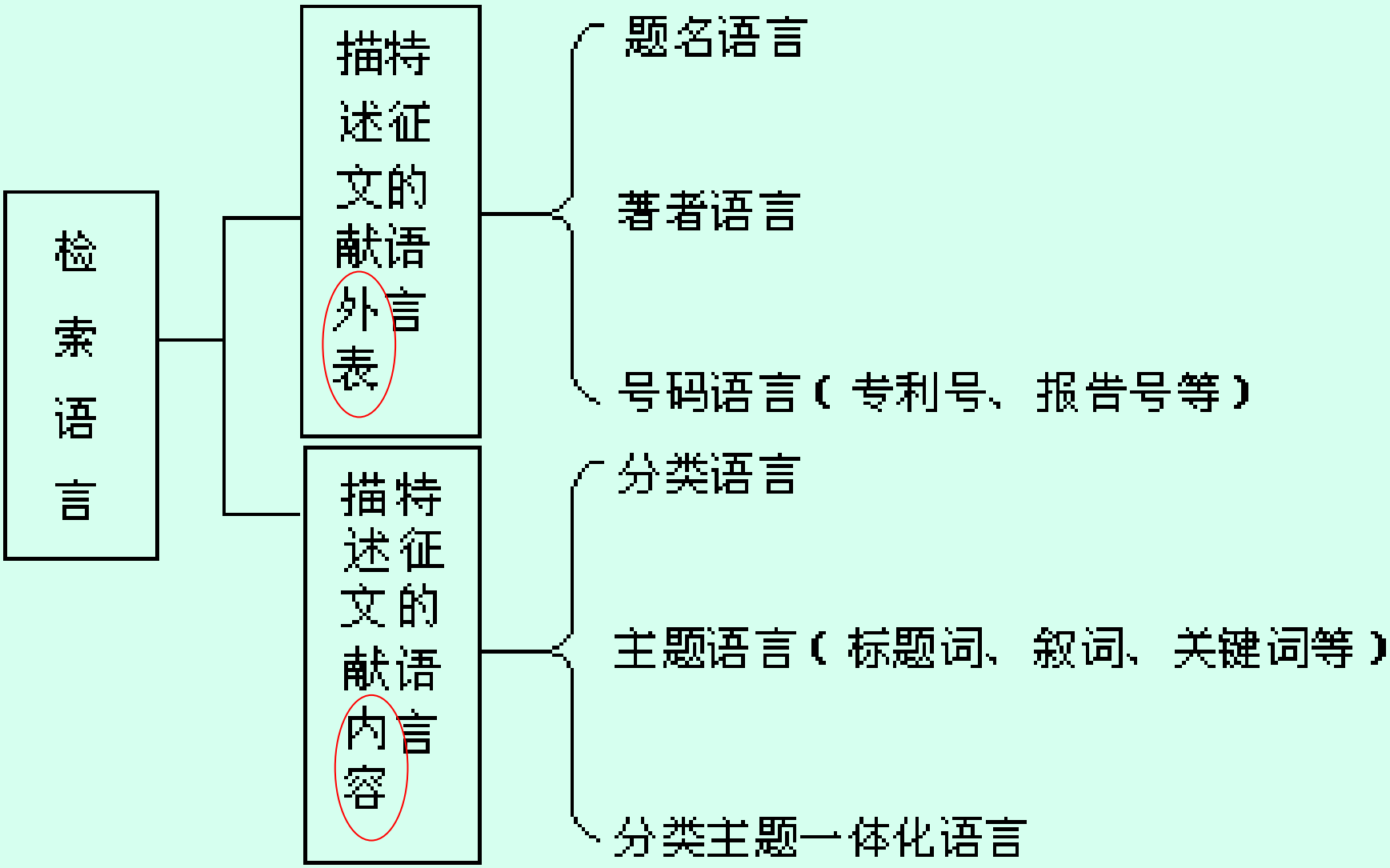


基于主题词的检索

基于元数据的检索

- 元数据是关于数据的数据
  - 是以计算机系统能够使用与处理的格式存在的与内容相关的数据，
  - 它是对内容的一种描述方式，表示内容的属性与结构信息。
- 元数据分为：
  - 描述元数据
  - 语义元数据
  - 控制元数据
  - 结构元数据

科技文献元数据检索语言



现代信息检索

- 基于内容检索
  - 基于内容的文本检索—全文检索
  - 基于内容的图像检索
  - 基于内容的视频检索
  - ……
- 结构化检索
- 浏览型检索

文本检索

- TR(Text Retrieval) 又：全文检索
- 文本检索的任务是根据用户的信息需求来定位满足用户需求的文档集。
- 与信息检索IR(Information Retrieval)的关系
  - 广义IR
    - 可以检索非文本信息(如:图象)
    - 多种任务(如:文本检索,过滤,分类,摘要)
  - 狭义IR = 文本检索
    - 给定文档集合
    - 给出信息检索需求
    - 检索相关文档

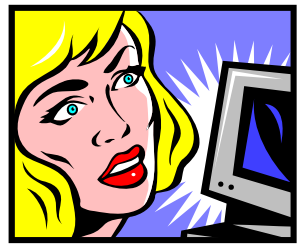
文本检索

- TR是IR中最重要的一项技术，
  - 大多数的信息需求是针对文本的。
  - 文本检索是信息检索的基础,包括很多具体应用，
    - 如：搜索引擎，数字图书馆，……
    - 检索, 定位, 筛选, 问答…
  - 文本检索的技术可以用来检索其他的媒体信息。

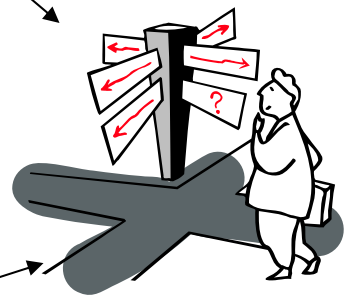
文本检索应用实例

The Web

Spider



User

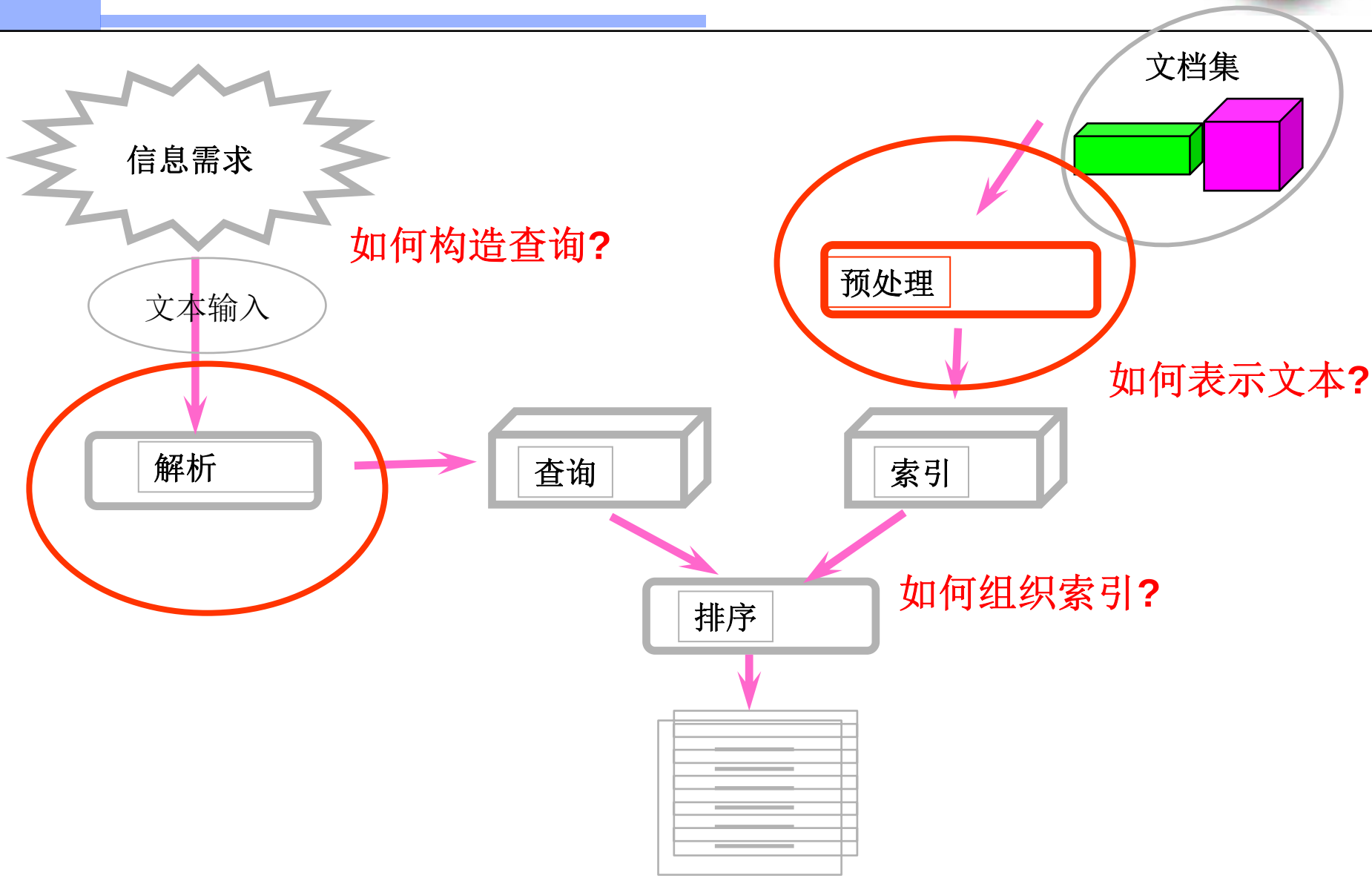


Inverted Index

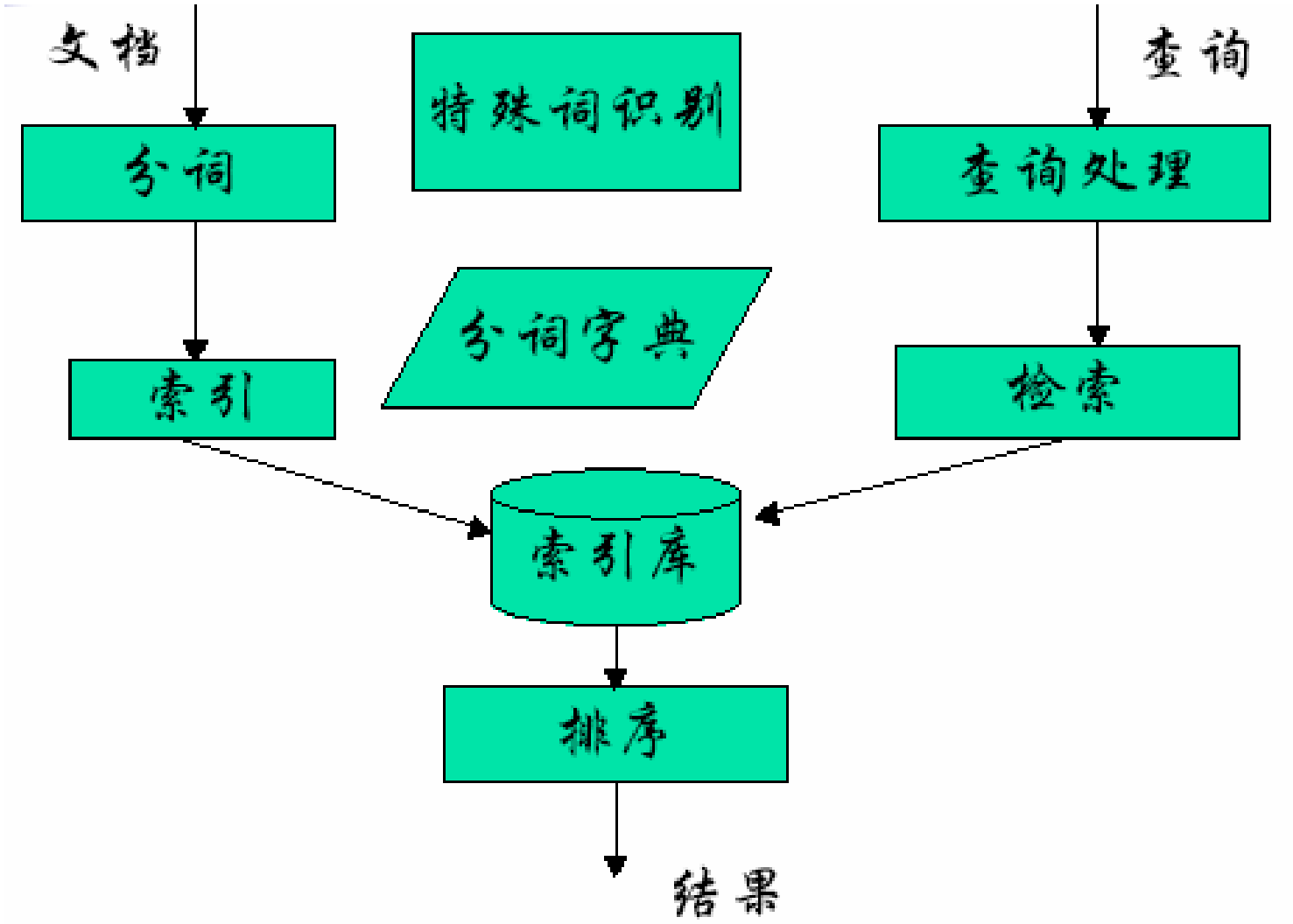
Library, etc.

Search Engine

文本检索过程



文本检索基本步骤



文本检索的难点

- 不同的用户有不同的检索需要
- 有时用户也不知道需要什么，或不知道怎样表达自己的需要
- 语言的歧义性
- 一词多义，多词一义
- 相关性的判定：80%

信息检索模型

信息检索模型

- **信息检索模型**（IR model），依照用户查询，对文档集合进行相关排序的一组前提假设和算法。
- IR模型可形式地表示为一个四元组

$$< D, Q, F, R(q_i,d_j) >$$

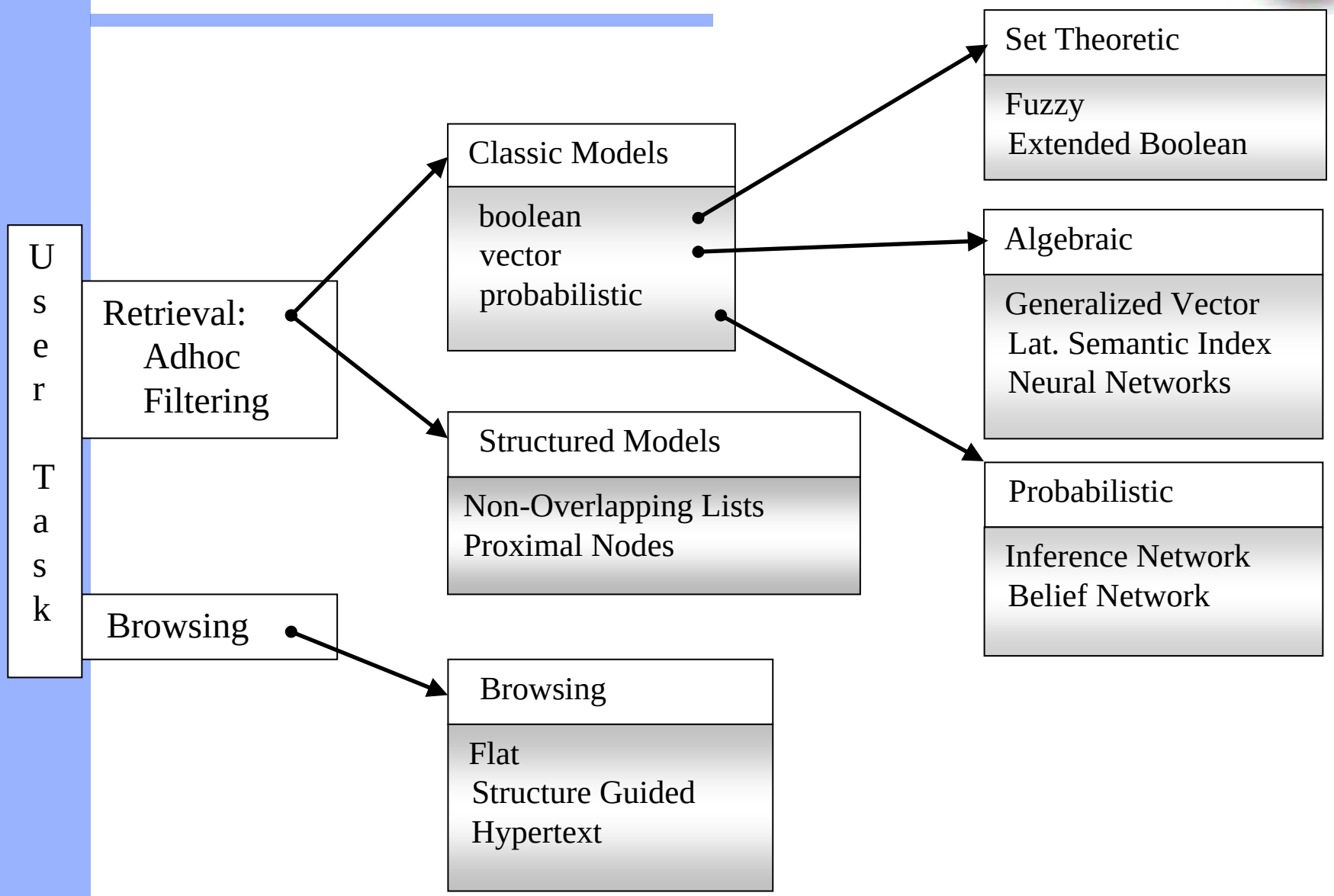
其中：

- D是一个文档集合，
- Q是一个查询集合，用户任务的表达
- F是一个框架，用以构建文档、查询以及它们之间关系的模型
- $R(q_i,d_j)$ 是一个排序函数，它给查询q_i和文档 d_j 之间的相关度赋予一个排序值

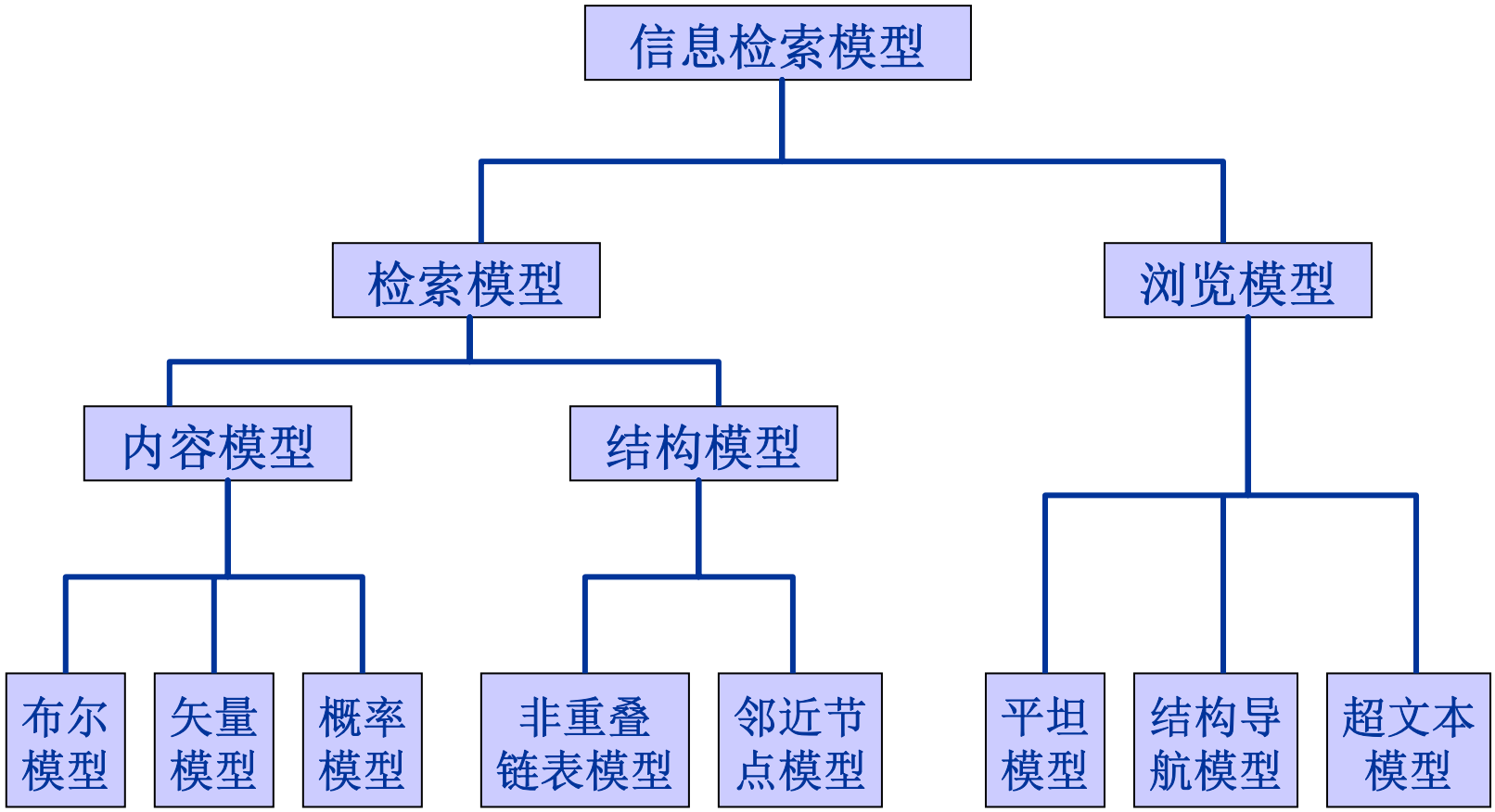
信息检索模型的分类

- 许多检索模型被提出，理论基础覆盖几何、逻辑、概率和统计
- 三大类：
  - 内容模型
  - 结构模型
  - 浏览模型

信息检索模型的分类


User Task
Retrieval:
Adhoc
Filtering
Browsing
Classic Models
boolean
vector
probabilistic
Structured Models
Non-Overlapping Lists
Proximal Nodes
Browsing
Flat
Structure Guided
Hypertext
Set Theoretic
Fuzzy
Extended Boolean
Algebraic
Generalized Vector
Lat. Semantic Index
Neural Networks
Probabilistic
Inference Network
Belief Network

信息检索模型的分类


信息检索模型
检索模型
浏览模型
内容模型
结构模型
布尔模型
矢量模型
概率模型
非重叠链表模型
邻近节点模型
平坦模型
结构导航模型
超文本模型

浏览模型

- 浏览模型概念
  - 用户的兴趣可能不在于提交一个对系统的查询。而是有意花一点时间来浏览文档空间，以寻找所关心的文档。
  - 用户是进行文档空间的浏览而不是搜索；
  - 浏览和搜索是不同的信息发现行为，通常来说，搜索比浏览更适合于有明确查找目标的用户。

浏览模型

- 浏览模型类别
  - 平坦浏览
    - 文档集可以被描述为平面上的点或是链表中的元素
    - 用户在这些文档上到处浏览，以寻找有关信息，
    - 在反馈过程中，用户通过在邻近文档中的浏览，查找出相关的资料，找出一些感兴趣的关键词

浏览模型

- 浏览模型类别
  - 结构导航
    - 文档被组织成为如目录那样的结构
    - 目录是类的层次结构
    - 对文档按照主题来分类和组织
    - →分类检索

浏览模型

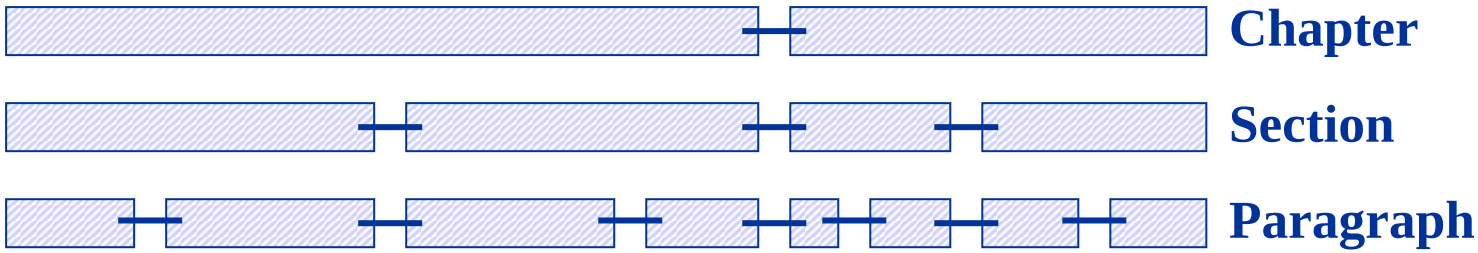
- 浏览模型类别
  - 超文本(Hypertext)
    - 超文本是一个允许以非顺序的方式在计算机上浏览文本的高层交互式导航结构
    - 它由节点和链组成，节点之间的关系由链表示，节点和链构成一个有向图。
    - 支持用户的非线性浏览和信息存取。
    - 超文本的导航过程可以被理解为一个有向图的游历过程，图中被链接的节点表示节点之间有某种语义关联。
    - →Web可看作一个巨大的超文本

结构模型

- 结构模型概念
  - 用户希望能够对文档中的某些结构组元中包含的信息进行检索
  - 例如，对出现在章节标题的词进行检索
  - 把文档内容与文档结构结合起来

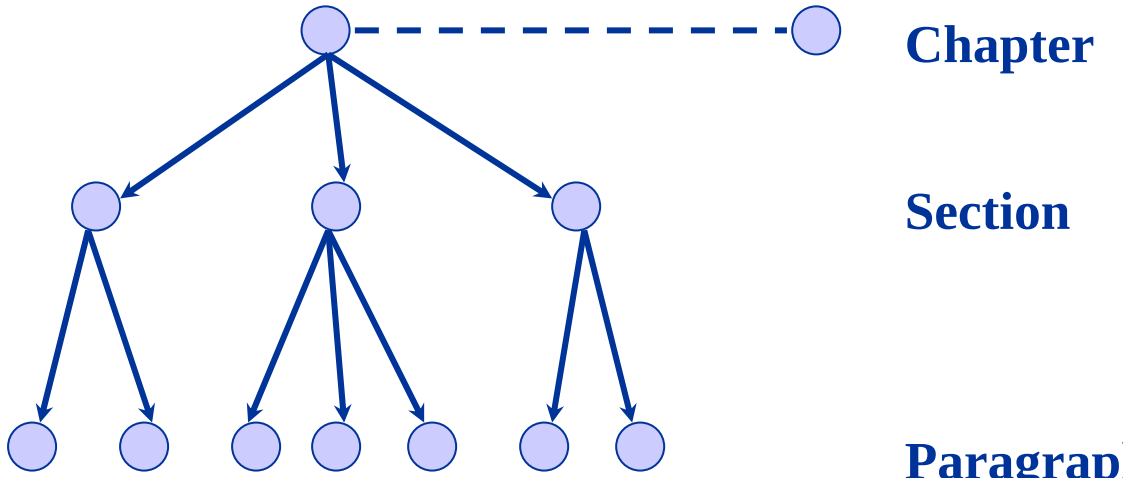
结构模型

- 结构模型类别
  - 非重叠链表模型
    - 文档中的整个文本划分为非重叠文本区域，并用链表连接起来
    - 相同链表中的文本区域没有重叠，而不同链表中的文本区域可能会重叠



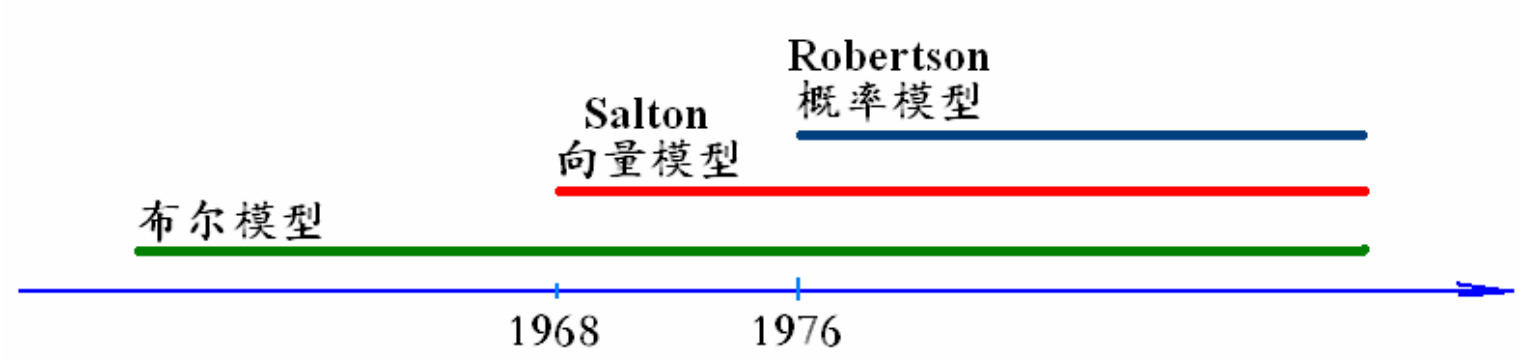
结构模型

- 结构模型类别
  - 邻近节点模型
    - 文档上定义一个或多个分层索引结构
    - 每个索引结构是一个严格的层次结构



内容模型

- 内容模型
  - 集合论模型：布尔模型、模糊集合模型、扩展布尔模型
  - 代数模型： 向量空间模型、广义向量空间模型、潜在语义标引模型、神经网络模型
  - 概率模型： 经典概率论模型、推理网络模型、置信（信念）网络模型、统计语言模型



布尔检索模型

- 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。
- 遵循两条基本规则： 每个索引词在一篇文档中只有两种状态：出现或不出现，对应权值为 0或1。
- 查询是由三种布尔逻辑运算符 and, or, not 连接索引词组成的布尔表达式。
  - 查询转化为一个主析取范式DNF

$$q = k_a \wedge (k_b \vee \neg k_c)$$

扩展的布尔模型

- Classical Boolean的最大缺点：只有0和1，没有ranking。要么返回大量结果，要么没有结果。
- Classical Boolean另一缺点：太僵化，在OR方式中，包含很多查询词的文档和包含少数词的文档是等同的；在AND方式中，即使缺少一个词，结果也是FALSE，等于一个词也没有
- 多种扩展模型
  - p-norm模型

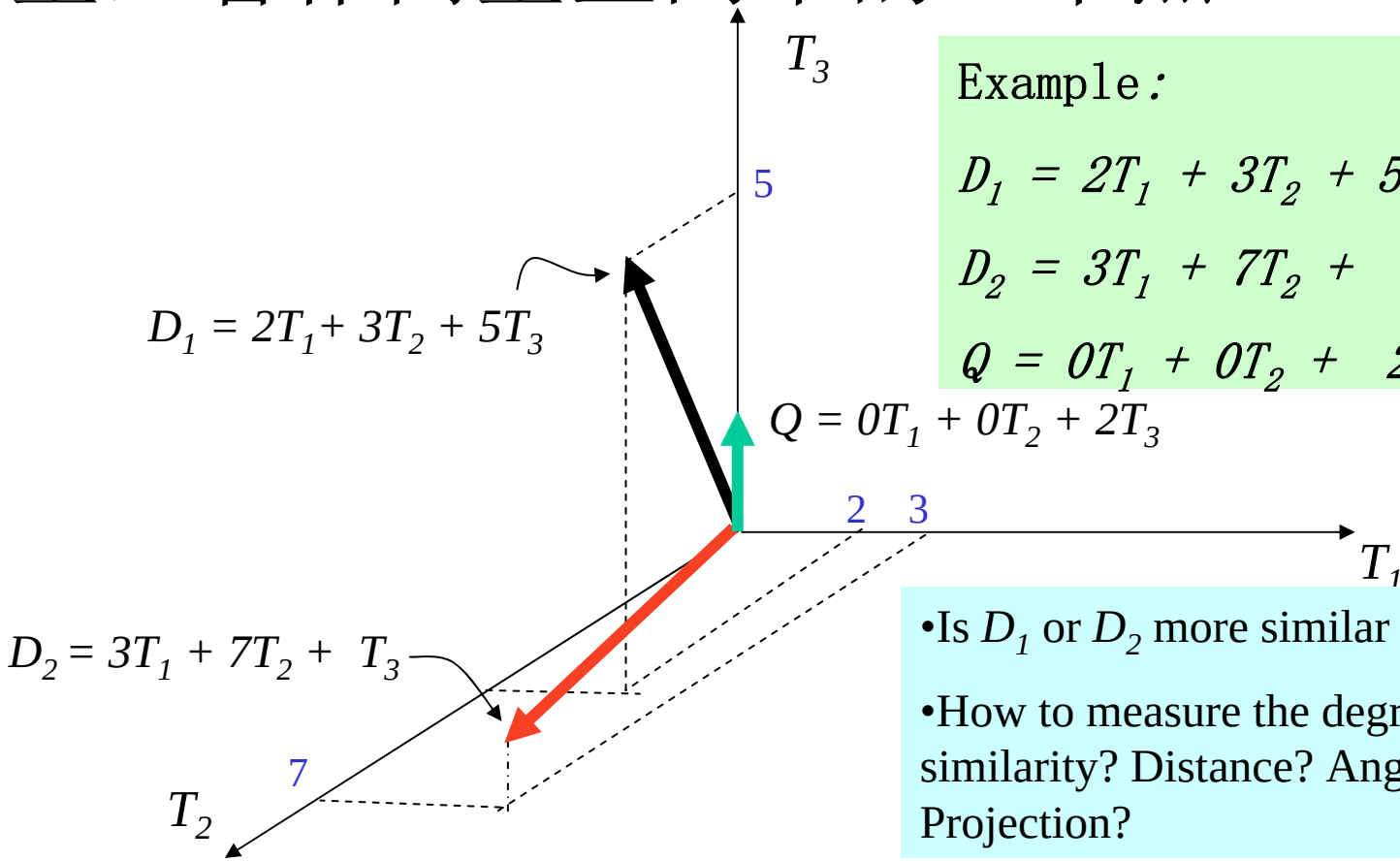
$$sim_{AND}(d,(t_1,W_{q1})AND...AND(t_n,W_{qn}))=1-\left(\frac{\sum_{i=1}^{n}((1-W_{di})^p W_{qi}{}^p)}{\sum_{i=1}^{n}W_{qi}{}^p}\right)^{\frac{1}{p}}$$

$$sim_{OR}(d,(t_1,W_{q1})OR...OR(t_n,W_{qn}))=\left(\frac{\sum_{i=1}^{n}(W_{di}{}^p W_{qi}{}^p)}{\sum_{i=1}^{n}W_{qi}{}^p}\right)^{\frac{1}{p}}$$

$$1\le p\le\infty$$

向量空间模型（VSM）

- 向量空间模型中将文档表达为一个矢量，看作向量空间中的一个点



Example:

$D_1 = 2T_1 + 3T_2 + 5T_3$

$D_2 = 3T_1 + 7T_2 + T_3$

$Q = 0T_1 + 0T_2 + 2T_3$

- Is D_1 or D_2 more similar to Q?
- How to measure the degree of similarity? Distance? Angle? Projection?

广义向量空间模型(GVSM)

- Generalized Vector Space Model
- 原理
  - 通过其在多个文档中出现的模式（occurrence patterns）来定义词项
  - 对查询中的词项也同样定义
  - 相似度的计算基于对d和q中重叠的模式来进行

广义向量空间模型(GVSM)

- 基本步骤
  - 将文档集合表达为一个向量

    C = [D1, D2, ..., Dm]
  - 将每一个词项按照其在文档集合上的分布也表达成一个向量：

    vec(ti)= [Tf(ti, D1), Tf(ti,D2), ..., Tf(ti, Dm)]
  - 定义词项之间的相似度：

    sim(ti, tj) = cos(vec(ti), vec(tj))

广义向量空间模型(GVSM)

- 基本步骤（续）
  - sim(q,d)不再是q和d的向量点乘，而是用上述“词项-词项”相似度的某个函数。
  - 例如，对q的每一个词项，分别得到它和d中词项的最大相似度，将这些最大相似度加起来得q和d的相似度

$$\mathbf{sim(q,d)} = \Sigma_i[max_j(sim(tq_i, td_j)]$$

广义向量空间模型(GVSM)

- 好处
  - 自动包含了部分相似的效果
    - “北大”和“创建一流”等就会较高的相似度（near-synonyms，其实是共生词）
  - 不需要做查询扩展或者相关性反馈
- 不利因素
  - 计算开销较大
  - 效果 = “向量空间 + Q扩展”的效果

概率模型

- 贝叶斯定理 $$P(A|B)=\frac{P(B|A)\cdot P(A)}{P(B)}$$
- 词条的独立假设：P(AB)= P(A) P(B) 当且仅当 A与B相互独立
- 由此对一篇文档而言，若文档中的各个词相互独立，则有 P(dj)=P(k1)…P(kt)

假设标引词独立，则（*Ki*：所有的词）

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

统计语言模型

- 一元(Unigram)模型
  - 假设词与词之间是相互独立的，一个词出现的概率与这个词前面的词没有存在必然联系。 $P(W_1W_2..W_n) = P(W_1)P(W_2)...P(W_n)$
- N元(NGram)模型
  - 假设词与词之间是相互关联的，一个词出现的概率与这个词前面的词存在一定的关联。
  - 二元(Bigram)模型

$$P(W_1W_2..W_n) = P(W_1)P(W_2 \mid W_1)...P(W_n \mid W_{n-1})$$

  - N元(NGram)模型

$$P(W_1W_2..W_n) = P(W_1)P(W_2 \mid W_1)..P(W_n \mid W_{n-1}W_{n-2}..W_1)$$

查询扩展

IR扩展技术

- 所有的匹配都是基于Term的匹配
- 相关扩展技术(用于查询,文档)
  - 词典资源的扩展
    - 增加相关词和同义词
  - 相关反馈
    - 增加一些“应该出现在查询中的词”
      - 用户参与
      - 直接的相关反馈
    - 自动伪相关反馈

查询扩展

- 较长的查询可能给带来好的查询结果
- 用户使用的查询词可能会和文档的用词不同

Q: *how to avoid heart disease*

D: "Factors in minimizing stroke and cardiac（心脏的）arrest（抑制）: Recommended dietary and exercise regimens（养生法）"

反馈学习技术

根据查询返回结果和用户反馈形成新的查询

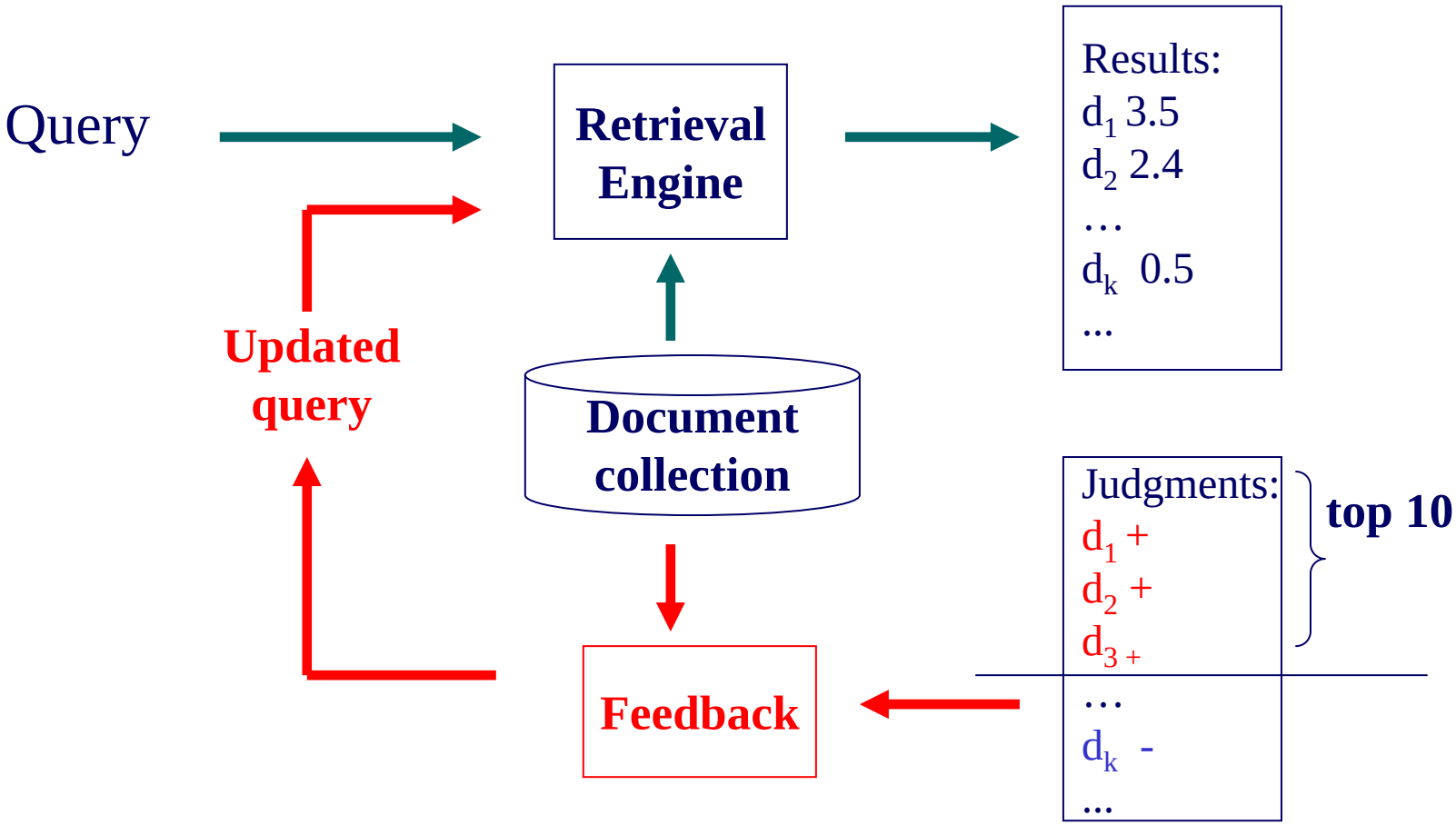
$Q' = F[Q, D_{ret}]$

F = 加权向量和，such as:

$$W(t,Q') = \alpha W(t,Q) + \beta W(t,D_{rel}) - \gamma W(t,D_{irr})$$

这里，t是Q中的成分

反馈学习技术


Query
Retrieval Engine
Results:
d_1 3.5
d_2 2.4
…
d_k 0.5
…
Updated query
Document collection
Judgments:
d_1 +
d_2 +
d_3 +
top 10
…
d_k -
…
Feedback

检索质量评价

检索质量的评价

- 文本检索的基本度量
  - {relevant}：与某查询相关的文档的集合。
  - {retrieved}：系统检索到的文档的集合。
  - {relevant} ∩ {retrieved}：既相关又被检索到的实际

检索质量的评价

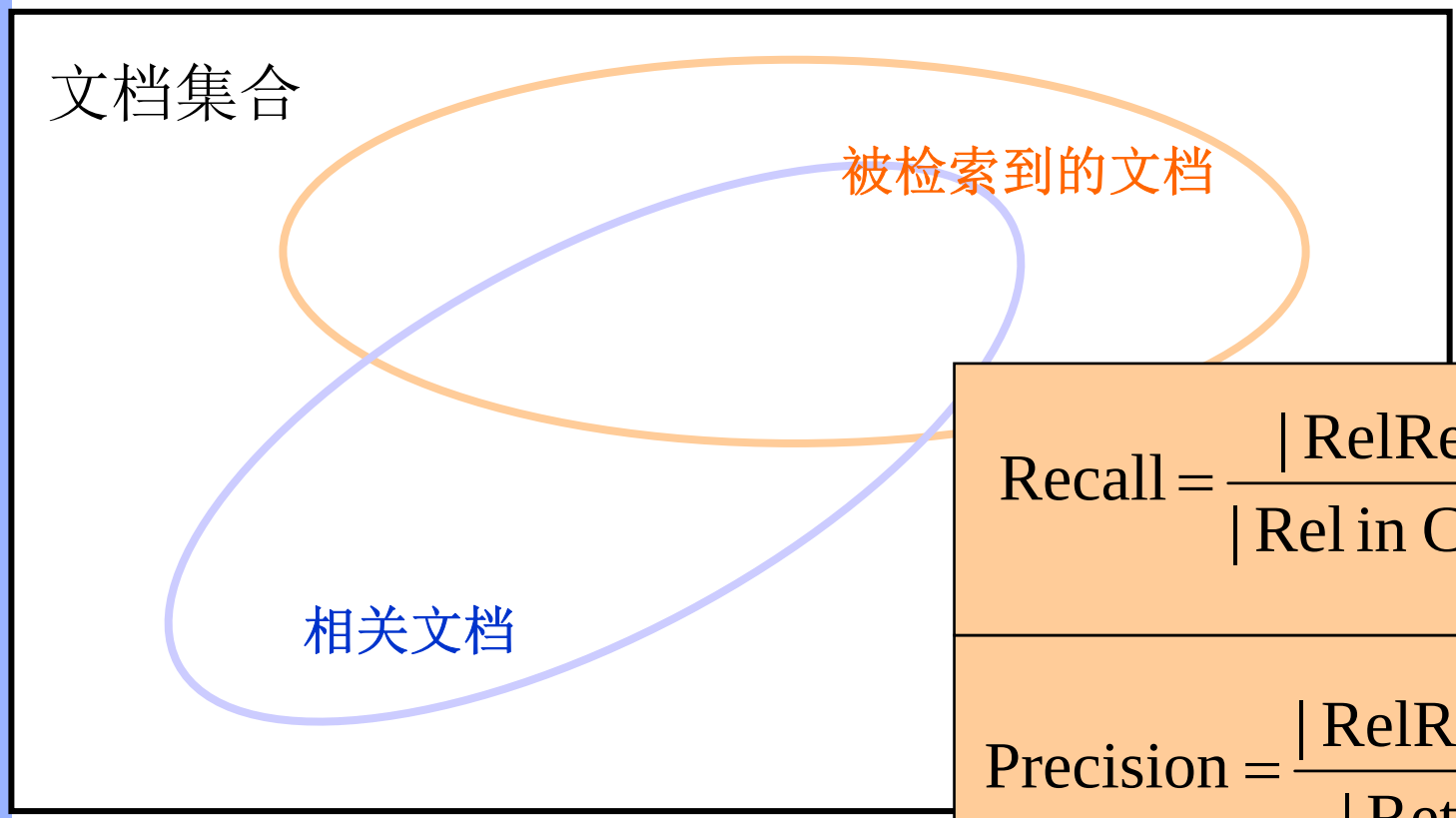
- 文本检索的基本度量
  - 查准率(precision)：既相关又被检索到的实际文档与被检索到的文档的百分比。

$$precision = \frac{|\{relevant\} \cap \{retrieved\}|}{|\{retrieved\}|}$$

  - 查全率(召回率recall)：既相关又被检索到的实际文档与查询相关的文档的百分比。

$$recall = \frac{|\{relevant\} \cap \{retrieved\}|}{|\{relevant\}|}$$

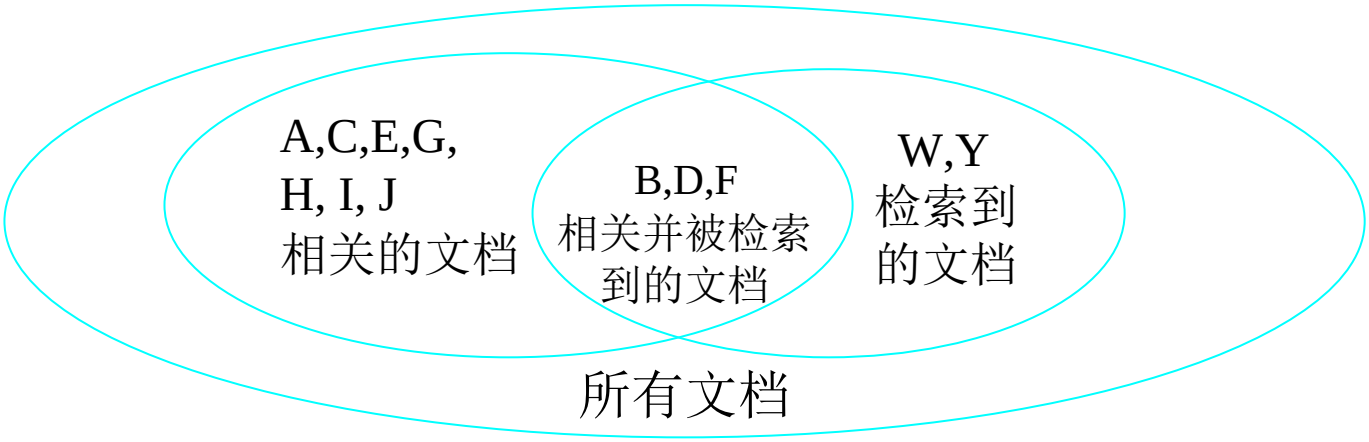
检索质量的评价



$$\text{Recall} = \frac{|\text{RelRetrieved}|}{|\text{Rel in Collection}|}$$

$$\text{Precision} = \frac{|\text{RelRetrieved}|}{|\text{Retrieved}|}$$

检索质量的评价

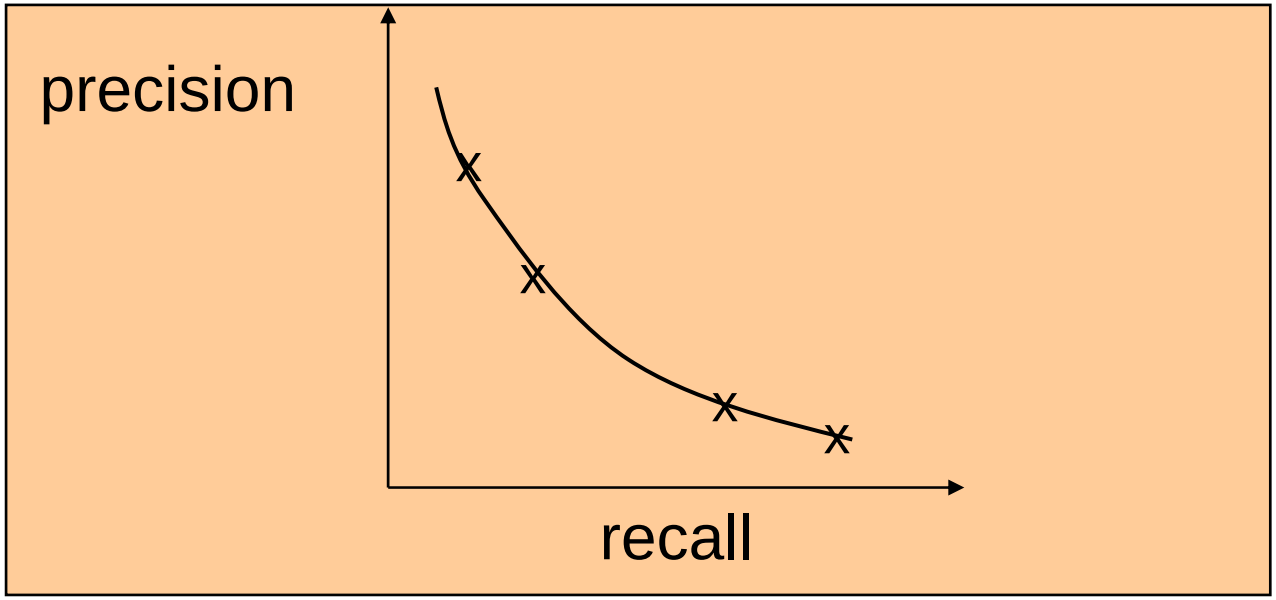


- {relevant} ={A, B, C, D, E, F, G, H, I, J} = 10
- {retrieved} = {B, D, F, W, Y} = 5
- {relevant} ∩ {retrieved} ={B, D, F} = 3
- 查准率：precision = 3/5 = 60%
- 查全率：recall = 3/10 = 30%
- 在同一检索模型中，分词决定准确性、索引算法决定速度

查准率/查全率曲线

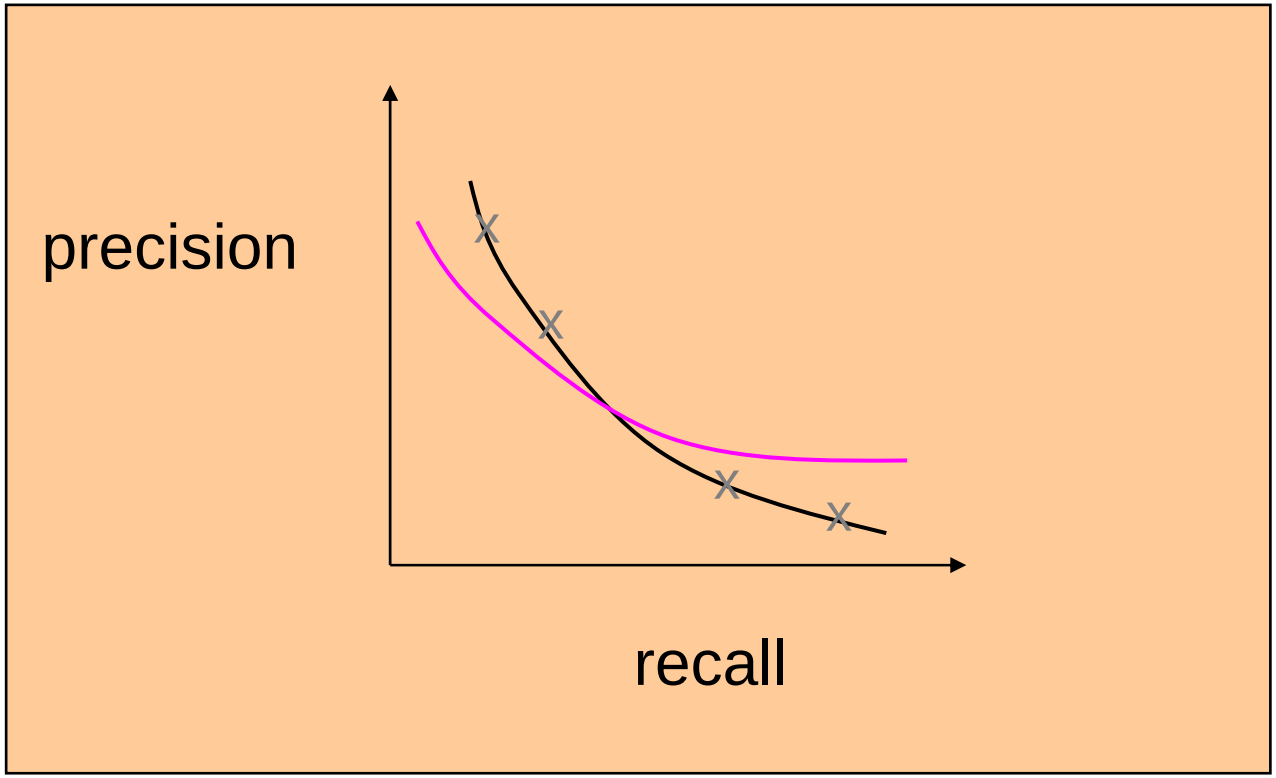
11-point precision curves

- 不同召回率下查准率的情况
- Plot precision at 10%, 20%, 30% ... , 100% recall
- IR system generates total ranking
- Note: this is an AVERAGE over MANY queries



查准率/查全率曲线

- Difficult to determine which of these two hypothetical results is better:



F-measure (F-度量)

- F-measure (F-度量)

$$F = \frac{1}{\alpha \frac{1}{P} + (1-\alpha)\frac{1}{R}}$$

  - α>0.5 : precision is more important
  - α<0.5 : recall is more important
  - Usually α=0.5

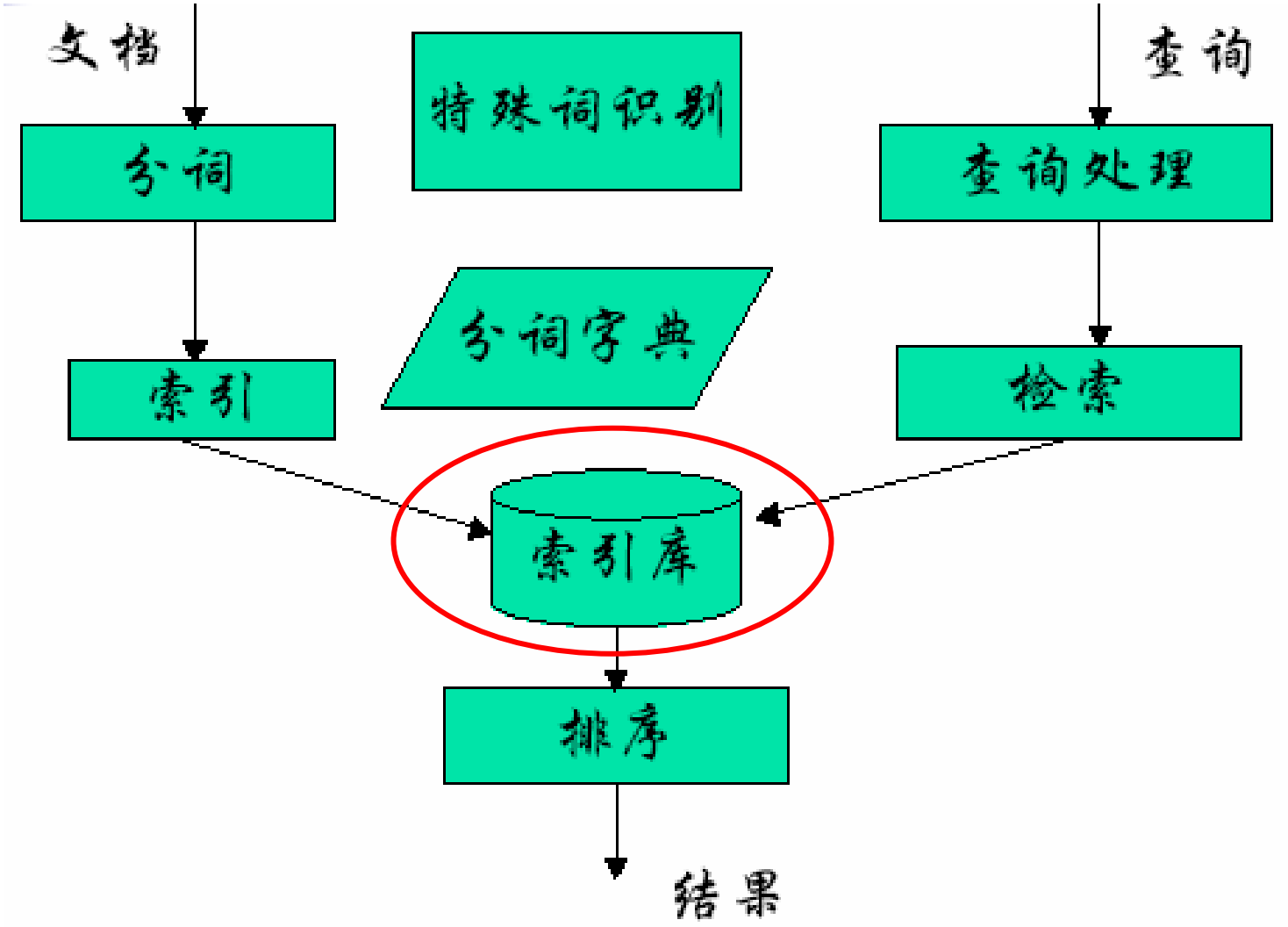
$$F = \frac{2PR}{P+R}$$

其他评价指标

- P@N：前N个结果的平均正确率
- MRR：第一个正确答案的平均序号倒数
- S@10：答案出现在前10个结果的查询比例

全文检索的索引技术

文本检索基本步骤



索引作用

- 索引
  - 提供从记录的特征快速查询到记录的数据结构(B树、散列表、位图索引等)
  - 数据库，文档数据库，SE/IR系统

数据库	结构化，查询和事务型更新
文档数据库	非结构化，查询和事务型更新
SE/IR系统	非结构化，查询

- 文本检索
  - 记录→文档doc，记录特征→索引词(index terms)

签名文件

- 定义：是一个记录每一个文档的特征的文件
- 方法：每一个特征对应一个固定长度的位串，一个比特位对应一个词汇，若某一位对应的词出现在文档中则该位置1，否则置0。
  - S_1

1	1	1	…	1

  - S_2

1	1	0	…	1

- 按位操作进行匹配，确定文档的相似性
- 可以多词对应一个比特位，来减少位串的长度，但增加搜索开销，存在多对一映射的缺点。

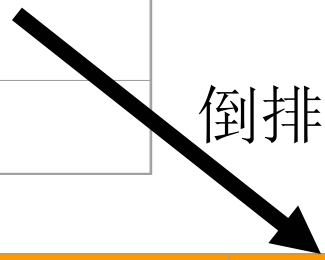
倒排索引

倒排

文档	内容
Doc1	….北京大学计算机系….
Doc2	….北京大学主页…..
Doc3	…计算机的发展…
。 。	

。

原始文档



倒排

倒排索引

索引词	索引项(posting list)
北京大学	<doc1><doc2>。。。
计算机	<doc1><doc3>。。。
。 。 。	。 。 。

倒排表

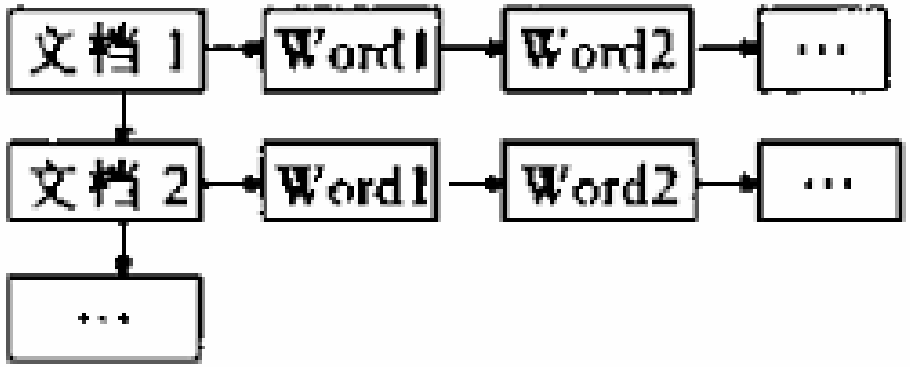


图 1　正排表结构

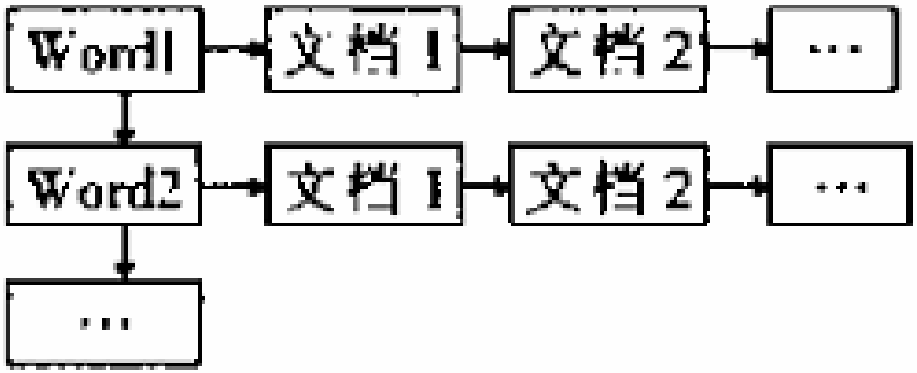


图 2　倒排表结构

倒排表例子

文档1

The quick brown fox jumped over the lazy dog's back.

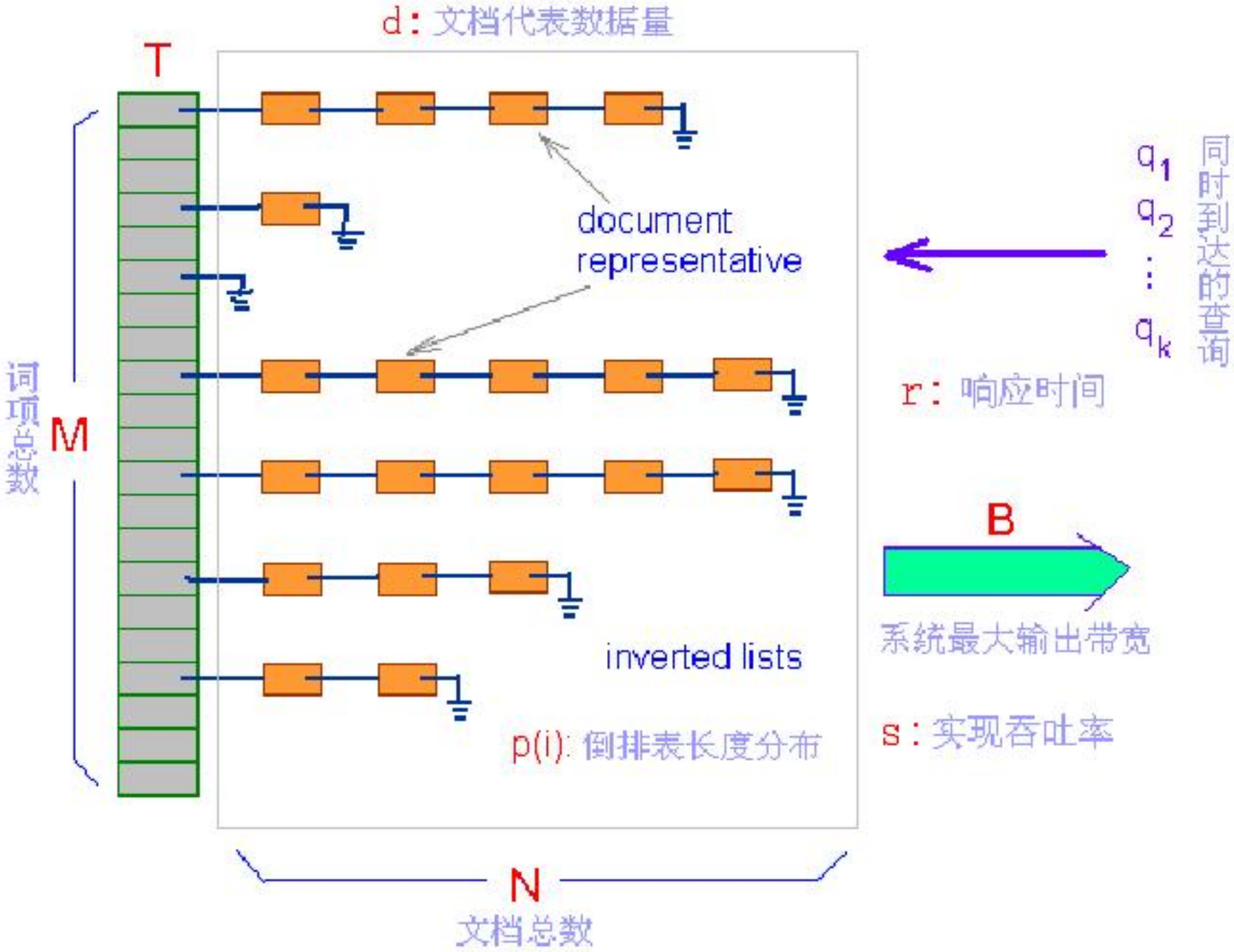
文档2

Now is the time for all good men to come to the aid of their party.

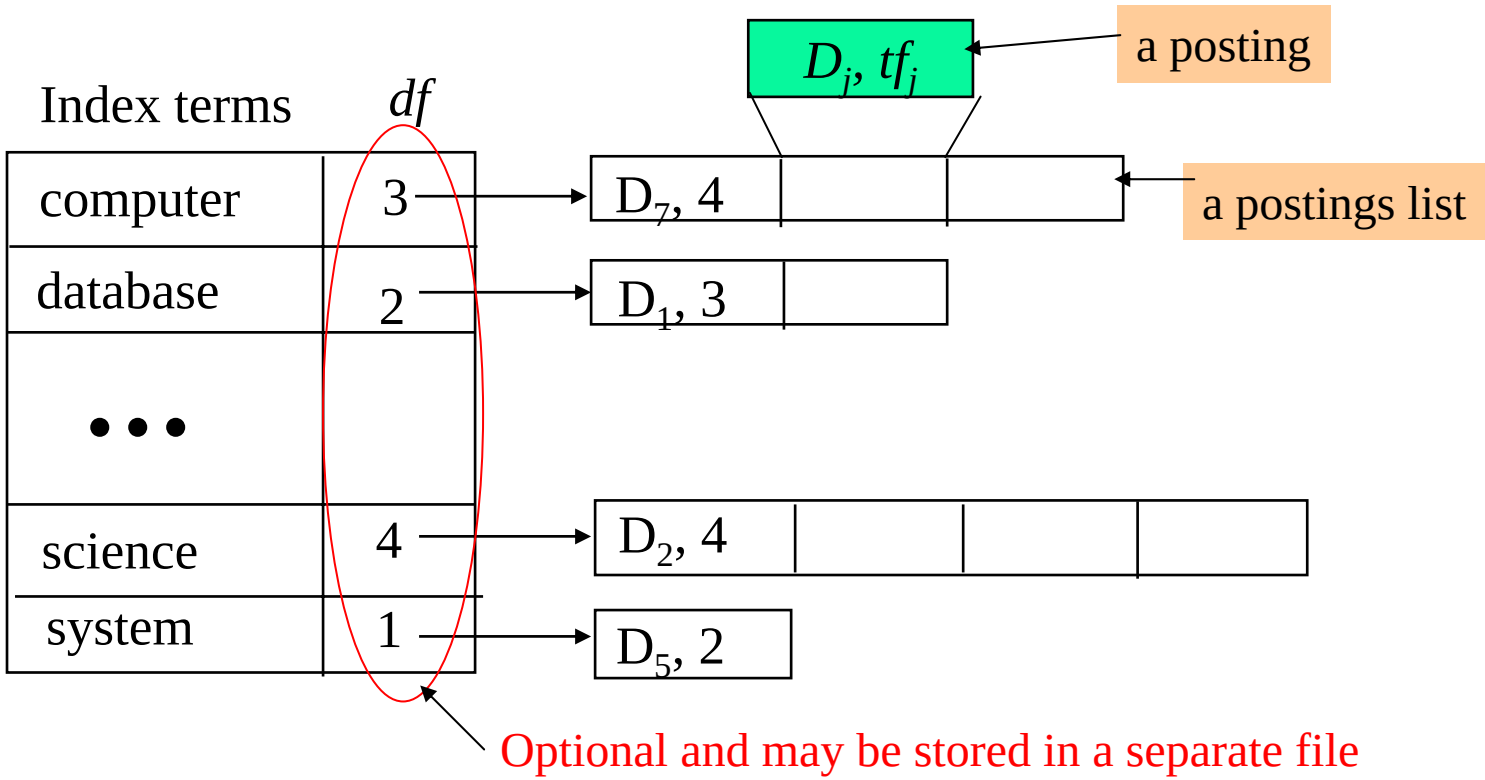
索引 Term	文档1	文档2
aid	0	1
all	0	1
back	1	0
brown	1	0
come	0	1
dog	1	0
fox	1	0
good	0	1
jump	1	0
lazy	1	0
men	0	1
now	0	1
over	1	0
party	0	1
quick	1	0
their	0	1
time	0	1

停用词表

for
is
of
's
the
to


d：文档代表数据量
T
document representative
词项总数
M
inverted lists
p(i)：倒排表长度分布
N
文档总数
q_1
q_2
q_k
同时到达的查询
r：响应时间
B
系统最大输出带宽
s：实现吞吐率

倒排表例子


Index terms
df
D_j, tf_j
a posting
computer
3
D_7, 4
a postings list
database
2
D_1, 3
• • •
science
4
D_2, 4
system
1
D_5, 2
Optional and may be stored in a separate file

索引单元

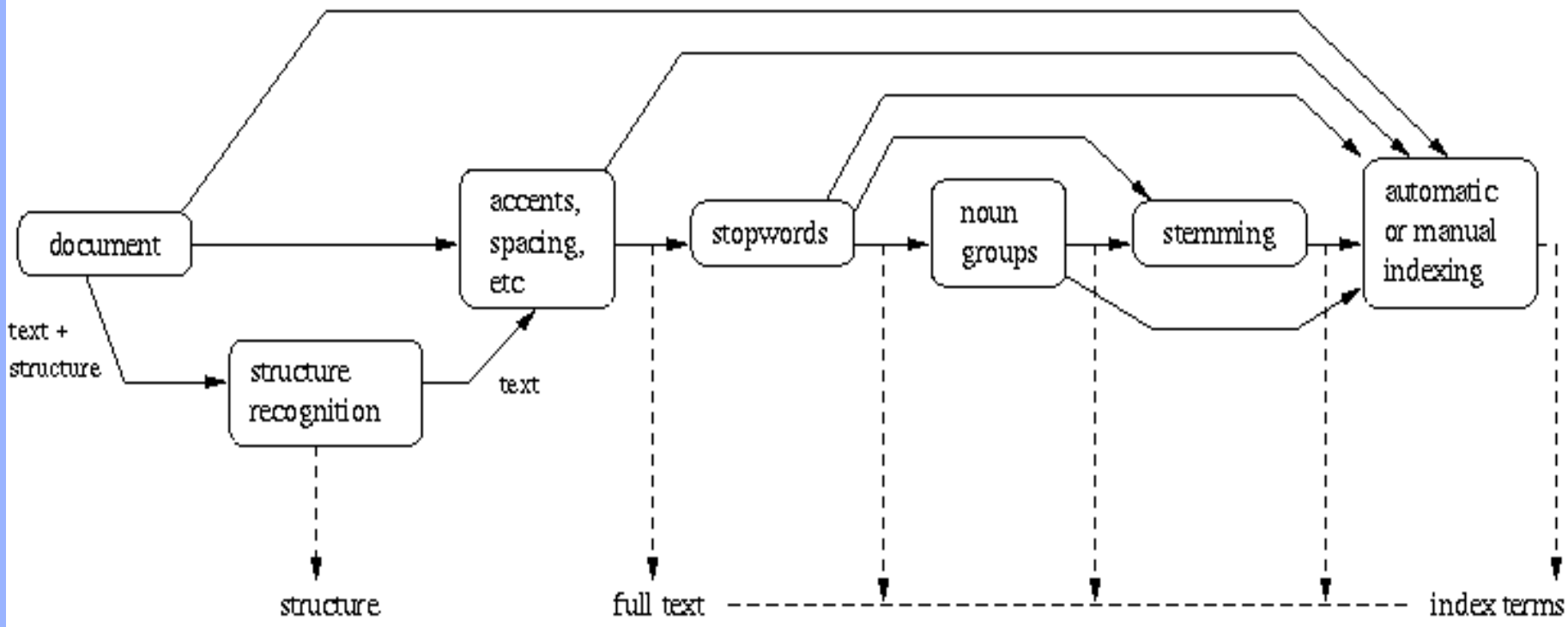
- 索引单元选择
  - Index term ≠ word
  - 理想：表达文档内容的语义单位
- 字索引
- 词索引：基于字或词的元组

  例句：小王准备做实验
  - 一元组：小王
  - 二元组：小王准备
  - N元组 ：小王准备做实验
  - 短语
- 混合索引

中文文本混合索引

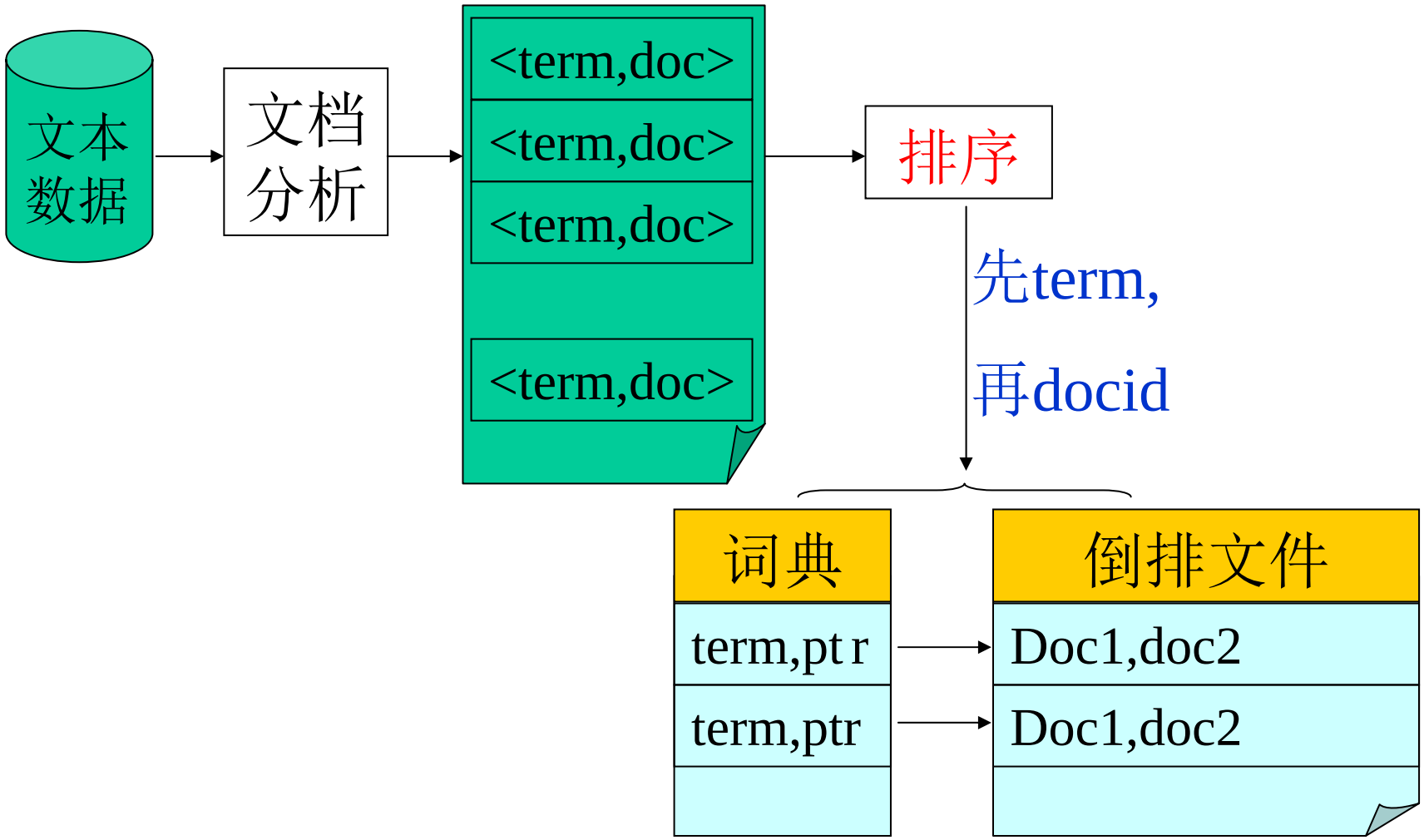
- 基本分词词典
  - 6万，选词较为严格
- 统计识别的未登录词扩展词典
  - 统计方法，精度不高
  - 如果加入到基本分词词典中，带来大量组合型歧义问题，不能正确处理。->混合索引
- 混合索引
  - 基本词典：“北京”“大学”，无“北京大学”；扩展词典：有“北京大学”
  - 文档中的“…北京大学…”，基本分词分为“北京”“大学”，扩展词典基础上再分为“北京大学”，索引按“北京”“大学”，“北京大学”这样三个单位建立。

文档索引单元提取


document
text +
structure
structure
recognition
structure
accents,
spacing,
etc
text
stopwords
noun
groups
stemming
automatic
or manual
indexing
full text
index terms

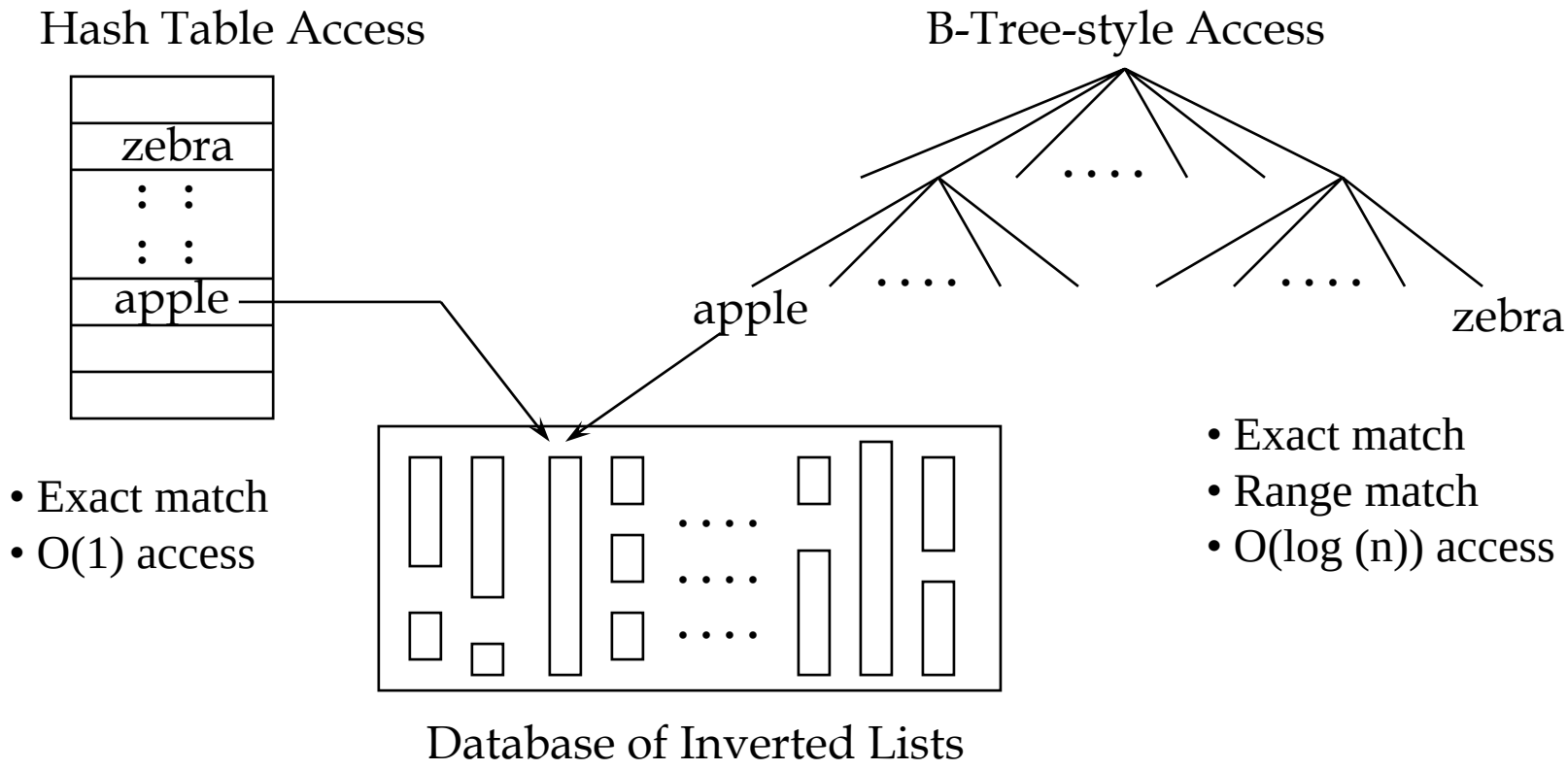
倒排索引创建

- 基本思想：排序



倒排索引创建

目标：快速定位索引单位所在的位置



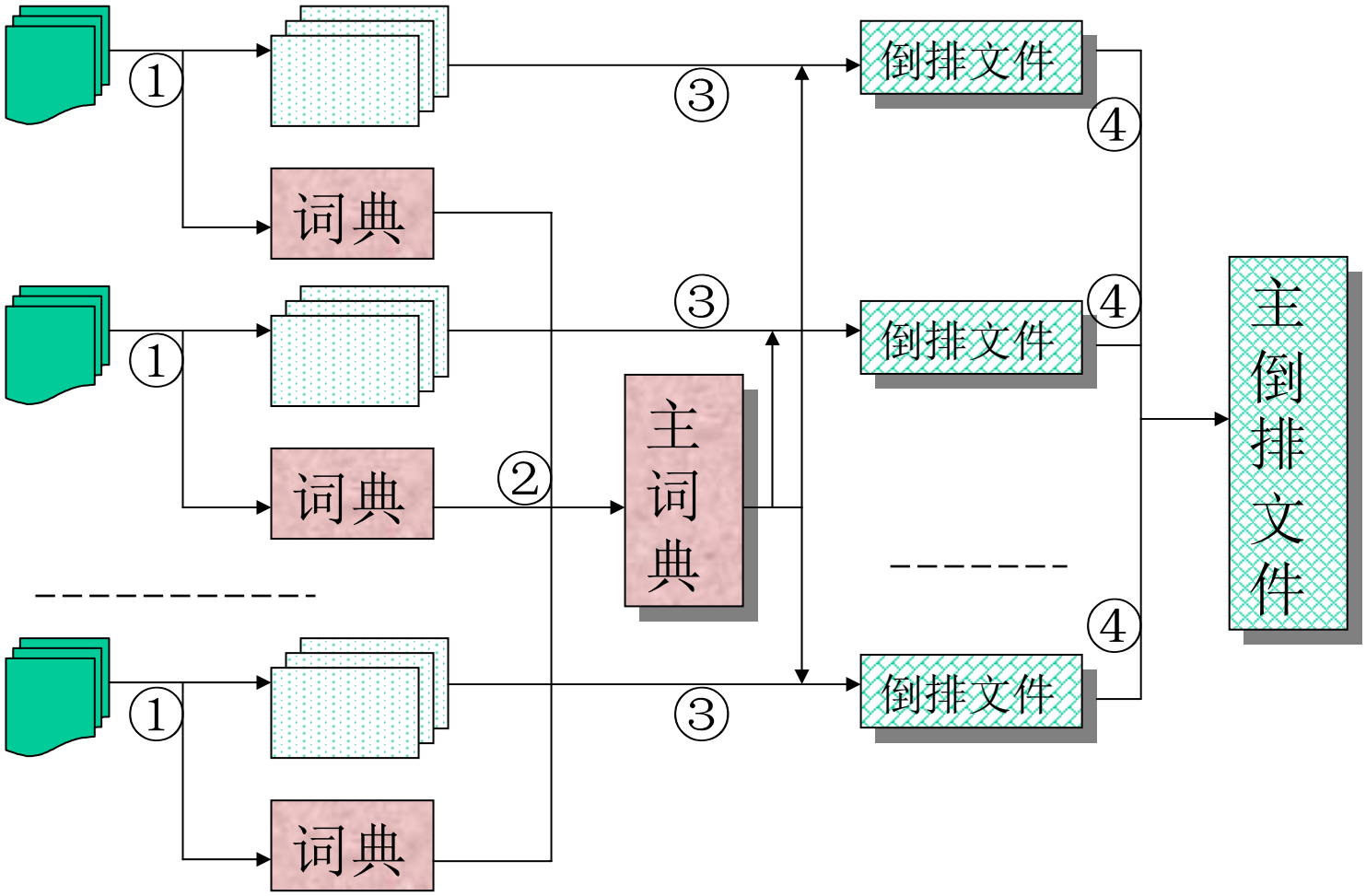
倒排索引创建—算法优化

- Term编码（词典组织）
  - 每个term用整数编码，减小存储空间
  - 英文—前缀编码
    - liber, liberal, liberalist…
  - 散列表(MPH, 无冲突散列)
- 减少磁盘的随机访问次数
  - <termid, docid>在内存中排序，排序结果分批写入磁盘，最后合并。
  - 两趟算法，在内存中直接倒排，小倒排文件分批写入磁盘，最后多路合并。
- 数据压缩

倒排索引创建—两趟算法

- 索引创建--两趟算法
  - 1.Parsing ,提取index term，统计df和tf，通过hash表转换为term id，生成词典文件(lexicon file)。
  - 2.按统计得到的index term的tf,df属性,可以估计出对应posting list长度，预申请空间。再次parsing文档集，在内存中执行倒排。结果保存到临时文件。
  - 3.对多次生成的临时倒排文件，多路合并，压缩输出，得到最终倒排文件。

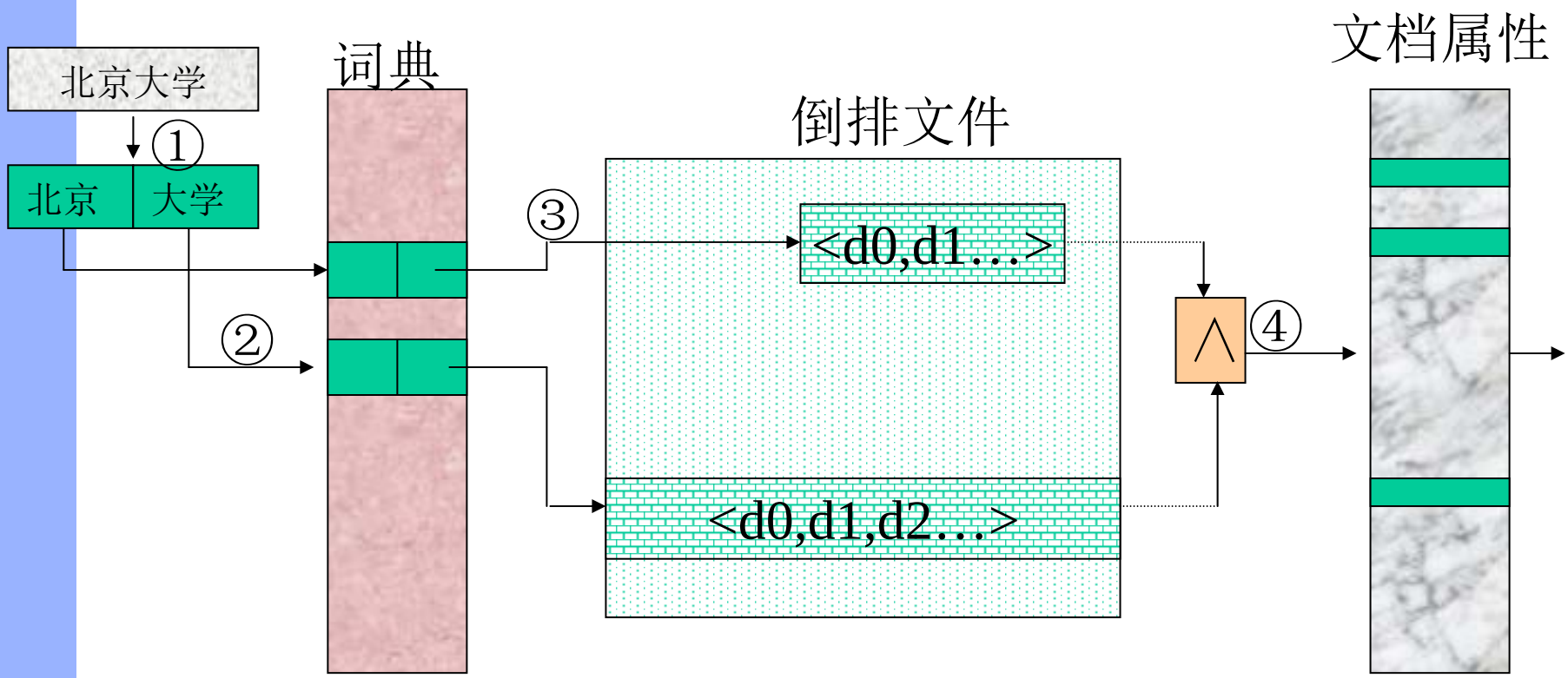
倒排索引创建—两趟算法



倒排索引创建—整数压缩

- 整数压缩
  - <docid, ….. >的整数序列压缩存贮。
  - 压缩的基本思想：→高频使用较短的位表示，低频使用较长的位表示（Huffman编码）→频率分布模式与编码
  - 有序整数序列，记录距离，改变频率分布模式，以提高压缩比
    - 1, 3, 7, 11, 13, 14 →1, 2, 4, 4, 2, 1
  - 编码方案
    - γ 系列、golomb系列、bytecode

索引查询



索引查询

- 布尔查询
  - 北京 AND 大学
- VSM rank查询
  - 相关度用文档相似度来计算
  - Similarity(Q,D)=COS(Q,D)

$$Cos(Q,D_d)=\frac{\sum_{t\in Q}W_{d,t}*W_{q,t}}{|W_q|\cdot|W_d|}=\frac{\sum_{t\in Q}f_{d,t}*idf*1*idf}{|W_q|\cdot|W_d|}=$$

$$=\frac{1}{|W_q|\cdot|W_d|}\sum_{t\in Q}f_{d,t}*\log(\frac{N}{ft})^2$$

索引查询

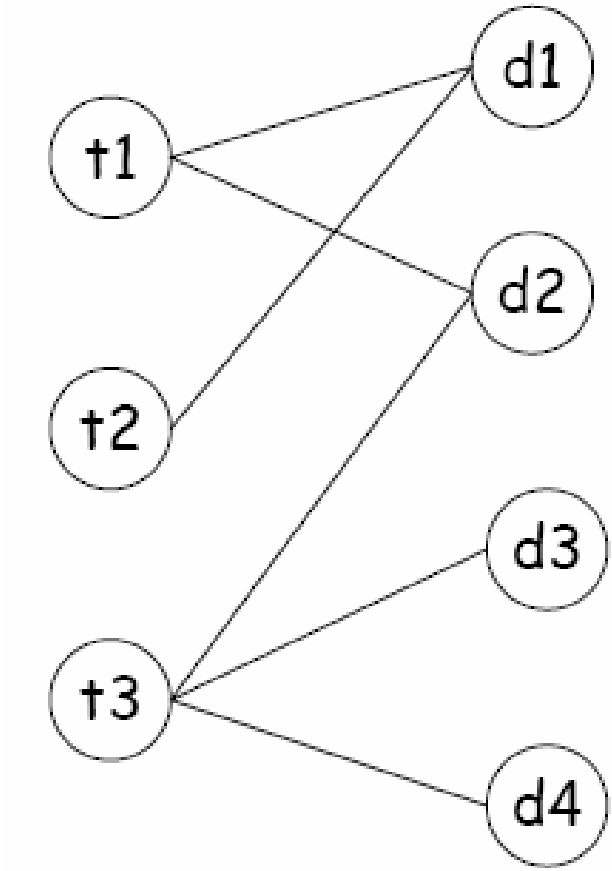
- VSM rank查询
  - Document-level索引：<docid,F(d,t)>
  - 增加文档属性数据库：|D|
- 短语、临近查询
  - 例如：“北京 网易” ，“北京大学”
  - Word-level索引：
    <docid,F(d,t),<loc1,loc2…locF(d,t)>>
- 结构查询
  - 例如：“北京大学 IN TITLE”
  - 在loc数据中用位标识记录 .VS. 在word-level index基础上使用text interval

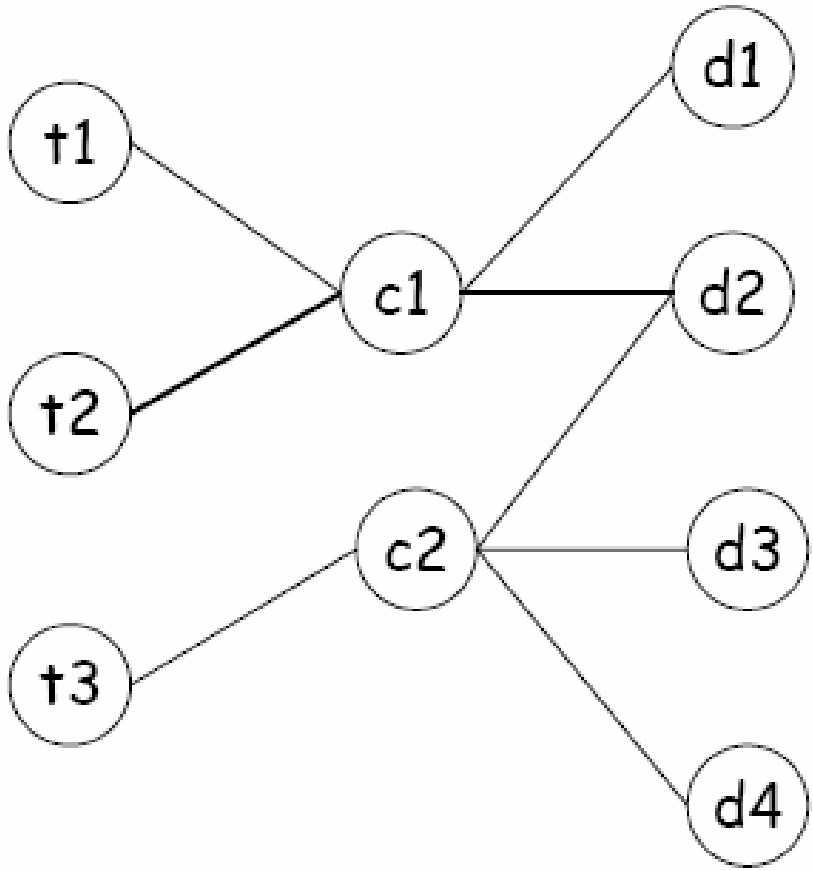
潜语义索引（LSI）
Latent Semantic Indexing

LSI?

- Concepts instead of words
- Mathematical model
  - relates documents and the concepts
- Looks for concepts in the documents
- Stores them in a concept space
  - related documents are connected to form a concept space
- Do not need an exact match for the query

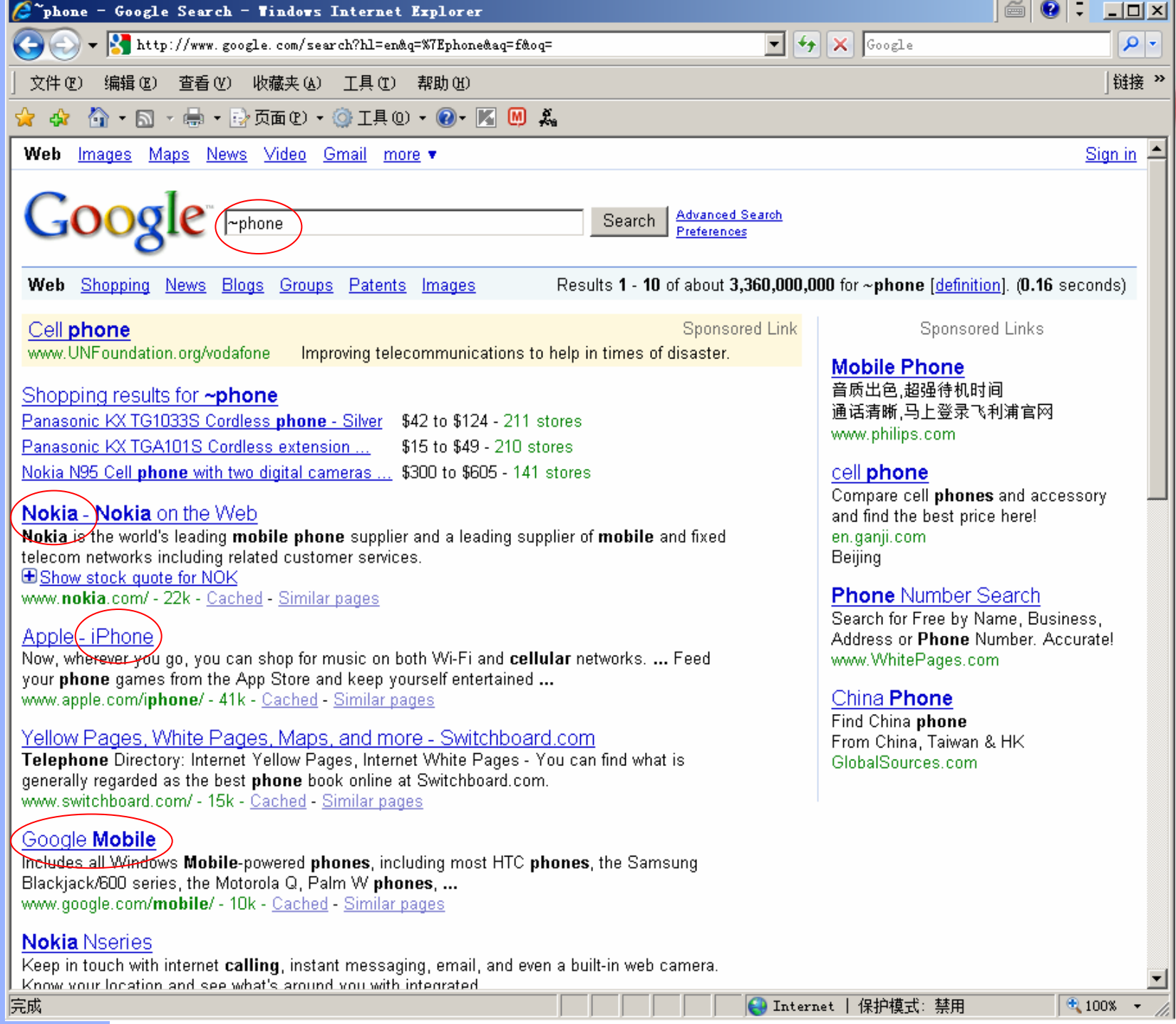
CONCEPTS

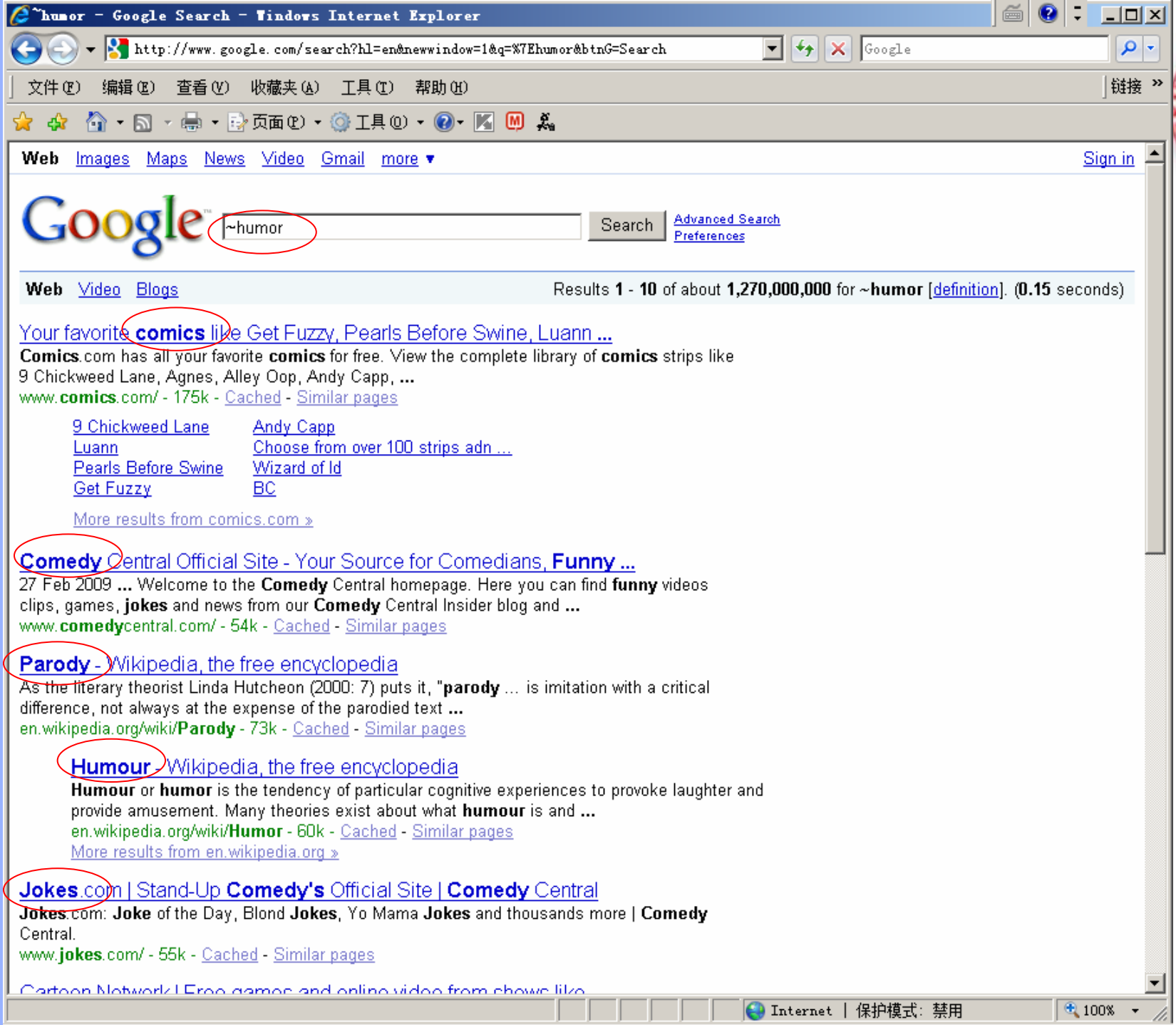




GOOGLE USES LSI

- Increasing its weight in ranking pages
  - ~ sign before the search term stands for the semantic search
  - “~phone”
    - the first link appearing is the page for “Nokia” although page does not contain the word “phone”
  - “~humor”
    - retrieved pages contain its synonyms; comedy, jokes, funny
- Google AdSense sandbox
  - check which advertisements google would put on your page


~phone - Google Search - Windows Internet Explorer
http://www.google.com/search?hl=en&q=%7Ephone&aq=f&oq=
文件(F) 编辑(E) 查看(V) 收藏夹(A) 工具(T) 帮助(H)
Web Images Maps News Video Gmail more ▾ Sign in
Google ~phone Search Advanced Search Preferences
Web Shopping News Blogs Groups Patents Images Results 1 - 10 of about 3,360,000,000 for ~phone [definition]. (0.16 seconds)
Cell phone Sponsored Link
www.UNFoundation.org/vodafone Improving telecommunications to help in times of disaster.
Shopping results for ~phone
Panasonic KX TG1033S Cordless phone - Silver $42 to $124 - 211 stores
Panasonic KX TGA101S Cordless extension ... $15 to $49 - 210 stores
Nokia N95 Cell phone with two digital cameras ... $300 to $605 - 141 stores
Nokia - Nokia on the Web
Nokia is the world's leading mobile phone supplier and a leading supplier of mobile and fixed telecom networks including related customer services.
Show stock quote for NOK
www.nokia.com/ - 22k - Cached - Similar pages
Apple - iPhone
Now, wherever you go, you can shop for music on both Wi-Fi and cellular networks. ... Feed your phone games from the App Store and keep yourself entertained ...
www.apple.com/iphone/ - 41k - Cached - Similar pages
Yellow Pages, White Pages, Maps, and more - Switchboard.com
Telephone Directory: Internet Yellow Pages, Internet White Pages - You can find what is generally regarded as the best phone book online at Switchboard.com.
www.switchboard.com/ - 15k - Cached - Similar pages
Google Mobile
Includes all Windows Mobile-powered phones, including most HTC phones, the Samsung Blackjack/600 series, the Motorola Q, Palm W phones, ...
www.google.com/mobile/ - 10k - Cached - Similar pages
Nokia Nseries
Keep in touch with internet calling, instant messaging, email, and even a built-in web camera.
Sponsored Links
Mobile Phone
音质出色,超强待机时间
通话清晰,马上登录飞利浦官网
www.philips.com
cell phone
Compare cell phones and accessory and find the best price here!
en.ganji.com
Beijing
Phone Number Search
Search for Free by Name, Business, Address or Phone Number. Accurate!
www.WhitePages.com
China Phone
Find China phone
From China, Taiwan & HK
GlobalSources.com
完成
Internet | 保护模式: 禁用
100%


~humor - Google Search - Windows Internet Explorer
http://www.google.com/search?hl=en&newwindow=1&q=%7Ehumor&btnG=Search
Google
文件(F) 编辑(E) 查看(V) 收藏夹(A) 工具(T) 帮助(H)
链接
页面(P) 工具(O)
Web Images Maps News Video Gmail more
Sign in
Google
~humor
Search
Advanced Search
Preferences
Web Video Blogs
Results 1 - 10 of about 1,270,000,000 for ~humor [definition]. (0.15 seconds)
Your favorite comics like Get Fuzzy, Pearls Before Swine, Luann ...
Comics.com has all your favorite comics for free. View the complete library of comics strips like 9 Chickweed Lane, Agnes, Alley Oop, Andy Capp, ...
www.comics.com/ - 175k - Cached - Similar pages
9 Chickweed Lane
Luann
Pearls Before Swine
Get Fuzzy
Andy Capp
Choose from over 100 strips adn ...
Wizard of Id
BC
More results from comics.com »
Comedy Central Official Site - Your Source for Comedians, Funny ...
27 Feb 2009 ... Welcome to the Comedy Central homepage. Here you can find funny videos clips, games, jokes and news from our Comedy Central Insider blog and ...
www.comedycentral.com/ - 54k - Cached - Similar pages
Parody - Wikipedia, the free encyclopedia
As the literary theorist Linda Hutcheon (2000: 7) puts it, "parody ... is imitation with a critical difference, not always at the expense of the parodied text ...
en.wikipedia.org/wiki/Parody - 73k - Cached - Similar pages
Humour - Wikipedia, the free encyclopedia
Humour or humor is the tendency of particular cognitive experiences to provoke laughter and provide amusement. Many theories exist about what humour is and ...
en.wikipedia.org/wiki/Humor - 60k - Cached - Similar pages
More results from en.wikipedia.org »
Jokes.com | Stand-Up Comedy's Official Site | Comedy Central
Jokes.com: Joke of the Day, Blond Jokes, Yo Mama Jokes and thousands more | Comedy Central.
www.jokes.com/ - 55k - Cached - Similar pages
Internet | 保护模式: 禁用
100%

HOW TO OBTAIN A CONCEPT SPACE?

- A set of documents
  - how to determine the similar ones?
    - examine the documents
    - try to find **concepts** in common
    - classify the documents
- LSI represents terms and documents in a high-dimensional space allowing *relationships between terms and documents to be exploited* during searching.
- Much simpler
  - use mathematical properties of the **term document matrix**,
    - i.e. determine the concepts by matrix computation.

HOW LSI WORKS?

- uses multidimensional vector space to place all documents and terms.
- Each dimension in that space corresponds to a concept existing in the collection.
- Thus underlying topics of the document is encoded in a vector.
- Common related terms in a document and query will pull document and query vector close to each other.

LSI: SVD for IR

- The entries of $(A^T A)_{n*n}$ may be interpreted as the pairwise documents in vector space.

$$A^T A = (USV^T)^T USV^T$$
$$= VS^T U^T USV^T$$
$$= VS^2 V^T$$

- Similarly, $(A\ A^T)_{m*m}$ as the pairwise term.

$$A\ A^T = US^2\ U^T$$

- The representations are vectors in an r-dimension subspace.

SVD for IR

After SVD, the dimension is reduced and

$$\mathrm{sim}(\boldsymbol{d}_i, \boldsymbol{d}_j) \approx \mathrm{sim}(\hat{\boldsymbol{d}}_i, \hat{\boldsymbol{d}}_j) \qquad (14)$$

Definition 9 The k-query vector is the projection of query vector q on $\mathrm{span}(\boldsymbol{U}_k)$, that is,

$$\begin{aligned}\hat{q} &= (q^T \bullet U_k \bullet S_k^{-1})^T \\ &= (\boldsymbol{q}^\top \boldsymbol{u}_1, \boldsymbol{q}^\top \boldsymbol{u}_2, \cdots, \boldsymbol{q}^\top \boldsymbol{u}_k)^\top\end{aligned}$$

Regard q as a special document vector, then by (14) the relativity between query and document is done.

Latent Semantic Kernels

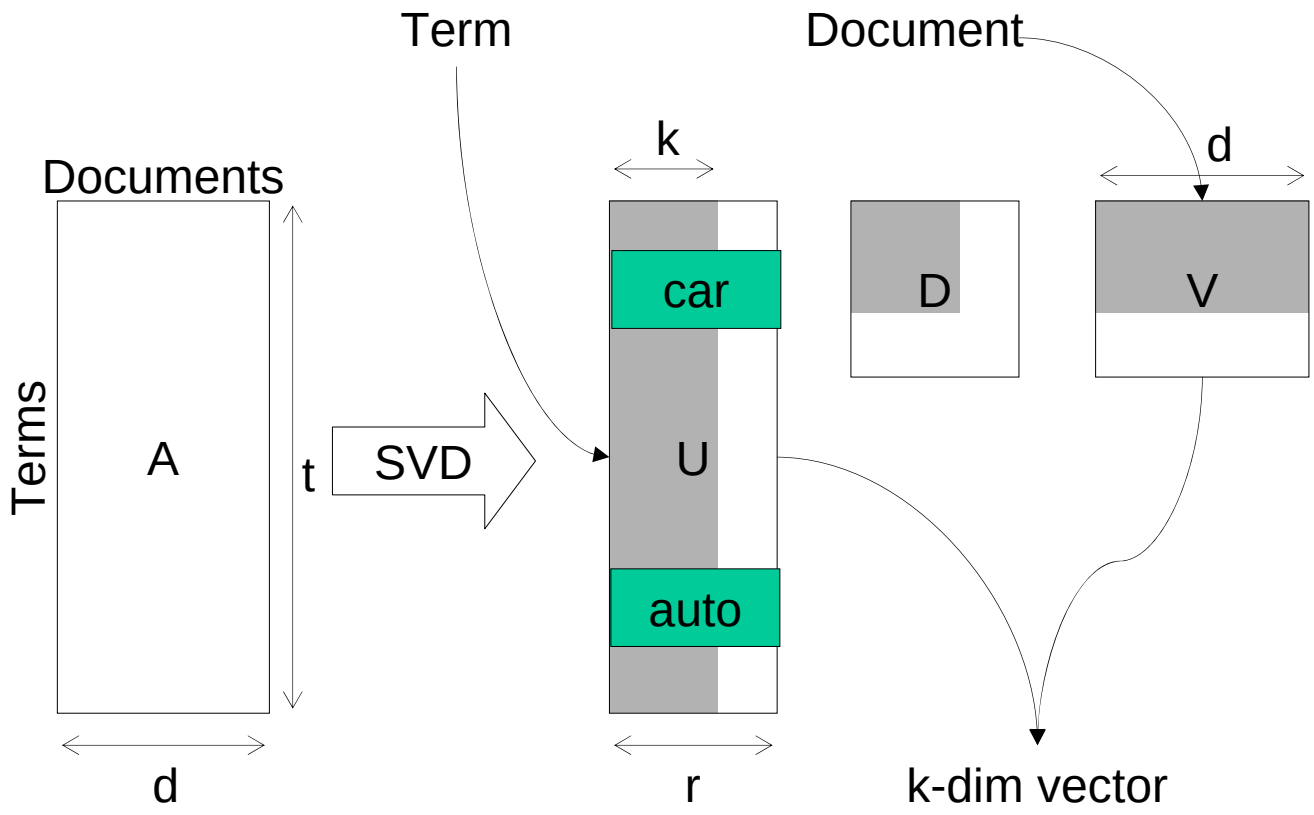
LSI uses a reduction of the first k columns of U.

$$d \mapsto \phi(d) U_k$$

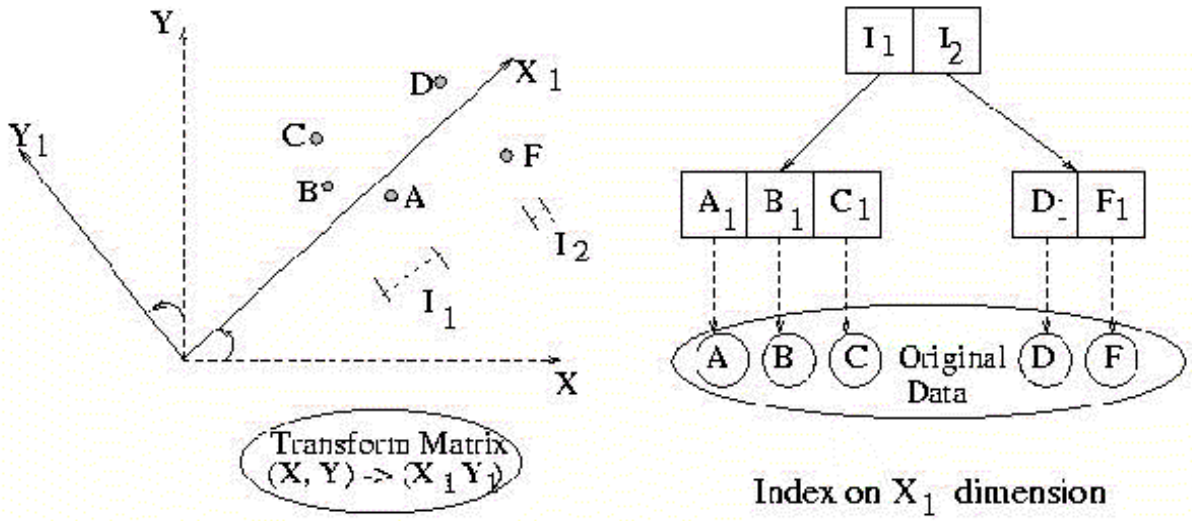
$$\tilde{\kappa}(d_1, d_2) = \sum_{i,j} \phi(d_1)_i U_k' U_k \phi(d_2)_j'$$

Latent semantic indexing

- SVD-变换与索引结构的结合



Latent semantic indexing



- 先进行SVD变换，将原始d-维数据变换到旋转后的空间
- 只保留前k维的值
- 用一维或多维索引结构（B^+树、R树等）进行索引

动态数据环境使用SVD

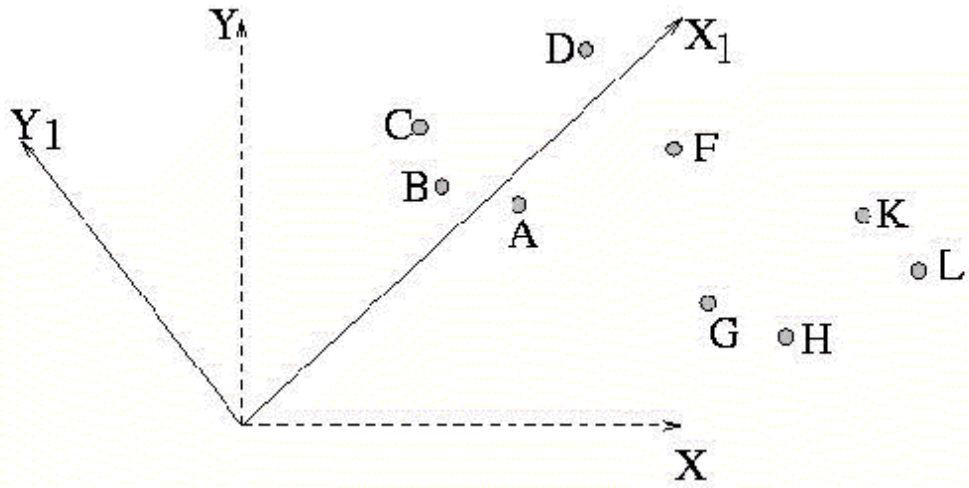
- 利用整个数据集进行变换，对静态数据效果很好
- 动态的数据库环境
  - 数据插入、删除、更新频繁，坐标轴要跟着旋转，以适应新的数据，否则性能下降
  - 要使性能不下降，SVD-变换需要重新计算。计算SVD-变换矩阵的时间复杂度为$O(n*d^2)$

动态数据环境使用SVD

- 每当数据更新时，都重新计算SVD-变换矩阵——查询精确度高，计算代价太大
- 不重新计算SVD-变换矩阵。——计算代价小，精确度随着更新数据的增多而下降
- 当数据更新引起的查询精度下降到某一阈值时，重新重新计算SVD-变换矩阵——牺牲一定的精确度，减小计算工作量

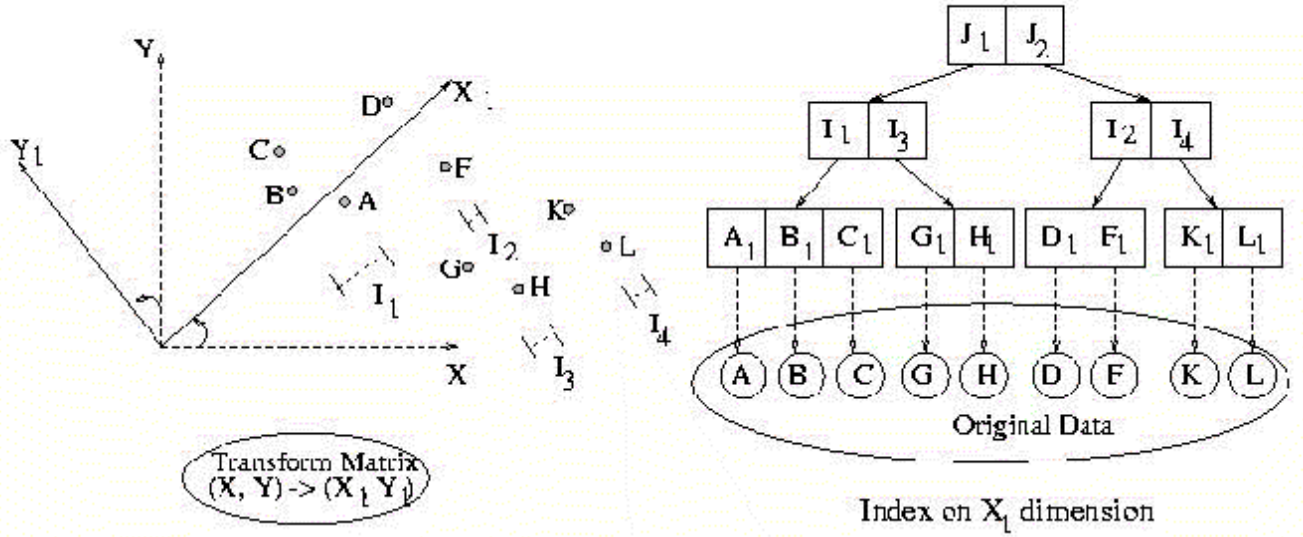
重新计算SVD-变换矩阵的方法

- 采用整个数据集进行重新计算
- 采用聚合数据进行重新计算



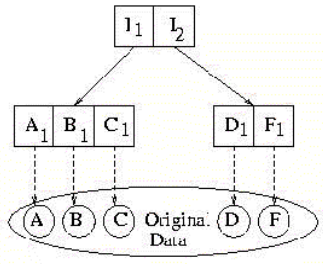
All-Data-SVD

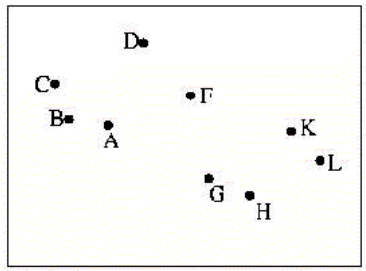
- 采用整个数据集进行重新计算
  - 第一步 数据访问：叶子结点A,B,…,L
  - 第二步 SVD计算：确定SVD-变换矩阵



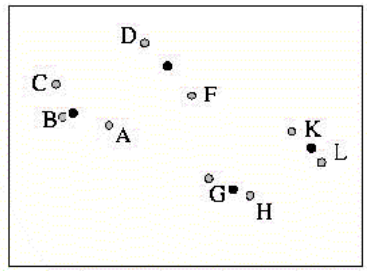
Approximation-SVD

- 采用聚合数据重新计算SVD-变换矩阵
  - 第一步　聚合数据集抽取：聚合数据集应能反映数据分布
    - 选择索引结构的某一层
      - 越靠近叶子层，精度越高，计算量越大
    - 对每个结点，计算该结点下所有数据的中心点
    - 所有的中心点组成聚合数据集
  - 第二步　SVD计算：用聚合数据计算SVD-变换矩阵





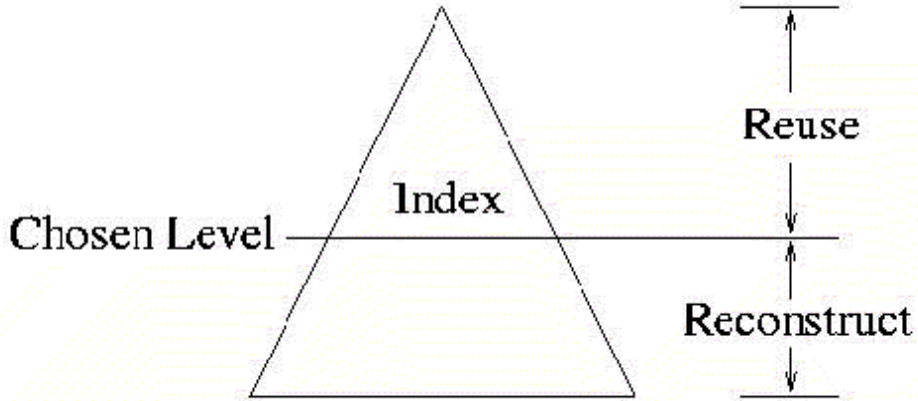
(a) All-Data-SVD



(b) Approximate-SVD

索引结构的更新

- 树重构（Tree-Reconstruct）
- 结构重用（Structure-Reuse）
- 重用-重构（ Reuse - Reconstruct）
  - 性能较好



实例分析

文档特征分析实例

文档集（9个文档）

c1 Human machine interface for Lab ABC computer application

c2 A survey of user opinion of computer system response time

c3 The EPS user interface management system

c4 System and human system engineering testing of EPS

c5 Relations of user-perceived response time to error measurement

m1 The generation of random, binary, unordered trees

m2 The intersection graph of paths in trees

m3 Graph minors IV: Widths of trees and well-quasi-ordering

m4 Graph minors: A survey

分词，选择特征词，过滤常用词

文档向量化

文档向量化（取12个特征词）

	c1	c2	c3	c4	c5	m1	m2	m3	m4
human	1	0	0	1	0	0	0	0	0
interface	1	0	1	0	0	0	0	0	0
computer	1	1	0	0	0	0	0	0	0
user	0	1	1	0	1	0	0	0	0
system	0	1	1	2	0	0	0	0	0
response	0	1	0	0	1	0	0	0	0
time	0	1	0	0	1	0	0	0	0
EPS	0	0	1	1	0	0	0	0	0
survey	0	1	0	0	0	0	0	0	1
trees	0	0	0	0	0	1	1	1	0
graph	0	0	0	0	0	0	1	1	1
minors	0	0	0	0	0	0	0	1	1

文档间相似度

- 文档间相似度
  - 点积

$$Sim(x,y) = x \bullet y = \sum_{k=1}^{t} (x_k \cdot y_k)$$

$$M = A^T \bullet A$$

$$M = ?$$

文档间相似度

- 文档间相似度
  - 点积

$$Sim(x,y) = x \bullet y = \sum_{k=1}^{t}(x_k \cdot y_k)$$

$$M = A^T \bullet A$$

  - 余弦

$$Sim(x,y) = \frac{x \bullet y}{|x| \cdot |y|} = \frac{\sum_{k=1}^{t}(x_k \cdot y_k)}{\sqrt{\sum_{k=1}^{t} x_k^{\ 2}} \cdot \sqrt{\sum_{k=1}^{t} \cdot y_k^{\ 2}}}$$

$$M'_{(i,j)} = M_{(i,j)} \Big/ \sqrt{M_{(i,i)} \cdot M_{(j,j)}}$$

文档间相似度（余弦）

$$Sim(x,y) = \frac{x \bullet y}{|x| \cdot |y|} = \frac{\sum_{k=1}^{t}(x_k \cdot y_k)}{\sqrt{\sum_{k=1}^{t} x_k^{\ 2}} \cdot \sqrt{\sum_{k=1}^{t} \cdot y_k^{\ 2}}}$$

	c1	c2	c3	c4	c5	m1	m2	M3	m4
c1	1	0.24	0.29	0.24	0	0	0	0	0
c2	0.24	1	0.41	0.33	0.71	0	0	0	0.24
c3	0.29	0.41	1	0.61	0.29	0	0	0	0
c4	0.24	0.33	0.61	1	0	0	0	0	0
c5	0	0.71	0.29	0	1	0	0	0	0
m1	0	0	0	0	0	1	0.71	0.58	0
m2	0	0	0	0	0	0.71	1	0.82	0.41
m3	0	0	0	0	0	0.58	0.82	1	0.67
m4	0	0.24	0	0	0	0	0.41	0.67	1

查询：相关度

Query: Human-Computer Interaction

$\mathbf{q}^T$= ⟨1 0 1 0 0 0 0 0 0 0 0 0⟩

$$Sim(x,y) = \frac{x \bullet y}{|x| \cdot |y|}$$

$$= \frac{\sum_{k=1}^{t}(x_k \cdot y_k)}{\sqrt{\sum_{k=1}^{t} x_k^{\ 2}} \cdot \sqrt{\sum_{k=1}^{t} \cdot y_k^{\ 2}}}$$

	c1	c2	c3	c4	c5	m1	m2	m3	m4
human	1	0	0	1	0	0	0	0	0
interface	1	0	1	0	0	0	0	0	0
computer	1	1	0	0	0	0	0	0	0
user	0	1	1	0	1	0	0	0	0
system	0	1	1	2	0	0	0	0	0
response	0	1	0	0	1	0	0	0	0
time	0	1	0	0	1	0	0	0	0
EPS	0	0	1	1	0	0	0	0	0
survey	0	1	0	0	0	0	0	0	1
trees	0	0	0	0	0	1	1	1	0
graph	0	0	0	0	0	0	1	1	1
minors	0	0	0	0	0	0	0	1	1

	c1	c2	c3	c4	c5	m1	m2	m3	m4
Query	**0.82**	**0.28**	0	**0.35**	0	0	0	0	0

查询结果

Query: Human-Computer Interaction

c1 Human machine interface for Lab ABC computer application

c2 A survey of user opinion of computer system response time

c3 The EPS user interface management system

c4 System and human system engineering testing of EPS

c5 Relations of user-perceived response time to error measurement

m1 The generation of random, binary, unordered trees

m2 The intersection graph of paths in trees

m3 Graph minors IV: Widths of trees and well-quasi-ordering

m4 Graph minors: A survey

	c1	c2	c3	c4	c5	m1	m2	m3	m4
Query	**0.82**	**0.28**	0	**0.35**	0	0	0	0	0

词频矩阵

词频矩阵（12 * 9）（12个特征维，9个文档）
(t=12, d=9)

A =

1	0	0	1	0	0	0	0	0
1	0	1	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0
0	1	1	0	1	0	0	0	0
0	1	1	2	0	0	0	0	0
0	1	0	0	1	0	0	0	0
0	1	0	0	1	0	0	0	0
0	0	1	1	0	0	0	0	0
0	1	0	0	0	0	0	0	1
0	0	0	0	0	1	1	1	0
0	0	0	0	0	0	1	1	1
0	0	0	0	0	0	0	1	1

SVD分解

$$A = USV^T$$

U =

(t=12, r=9 (r≤min(t , d))

0.22	-0.11	0.29	-0.41	-0.11	-0.34	0.52	-0.06	-0.41
0.20	-0.07	0.14	-0.55	0.28	0.50	-0.07	-0.01	-0.11
0.24	0.04	-0.16	-0.59	-0.11	-0.25	-0.30	0.06	0.49
0.40	0.06	-0.34	0.10	0.33	0.38	0.00	0.00	0.01
0.64	-0.17	0.36	0.33	-0.16	-0.21	-0.17	0.03	0.27
0.27	0.11	-0.43	0.07	0.08	-0.17	0.28	-0.02	-0.05
0.27	0.11	-0.43	0.07	0.08	-0.17	0.28	-0.02	-0.05
0.30	-0.14	0.33	0.19	0.11	0.27	0.03	-0.02	-0.17
0.21	0.27	-0.18	-0.03	-0.54	0.08	-0.47	-0.04	-0.58
0.01	0.49	0.23	0.03	0.59	-0.39	-0.29	0.25	-0.23
0.04	0.62	0.22	0.00	-0.07	0.11	0.16	-0.68	0.23
0.03	0.45	0.14	-0.01	-0.30	0.28	0.34	0.68	0.18

SVD分解

$$A = USV^T$$

S = (r * r, r=9 (r≤min(t ,d))

3.34								
	2.54							
		2.35						
			1.64					
				1.50				
					1.31			
						0.85		
							0.56	
								0.36

SVD分解

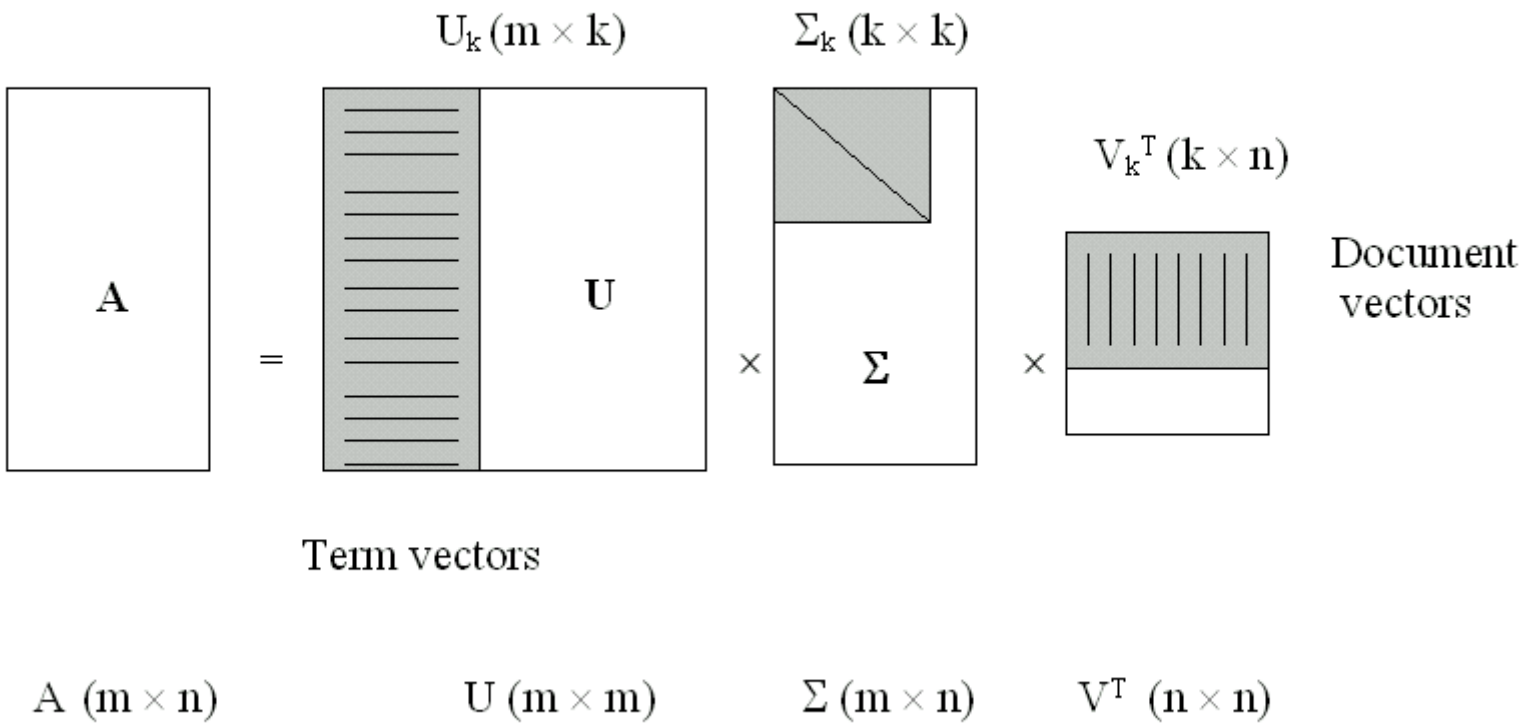
$$A = USV^T$$

V =

V:d * r

0.20	-0.06	0.11	-0.95	0.05	-0.08	0.18	-0.01	-0.06
0.61	0.17	-0.50	-0.03	-0.21	-0.26	-0.43	0.05	0.24
0.46	-0.13	0.21	0.04	0.38	0.72	-0.24	0.01	0.02
0.54	-0.23	0.57	0.27	-0.21	-0.37	0.26	-0.02	-0.08
0.28	0.11	-0.51	0.15	0.33	0.03	0.67	-0.06	-0.26
0.00	0.19	0.10	0.02	0.39	-0.30	-0.34	0.45	-0.62
0.01	0.44	0.19	0.02	0.35	-0.21	-0.15	-0.76	0.02
0.02	0.62	0.25	0.01	0.15	0.00	0.25	0.45	0.52
0.08	0.53	0.08	-0.03	-0.60	0.36	0.04	-0.07	-0.45

SVD分解降维


U_k (m × k)
Σ_k (k × k)
V_k^T (k × n)
A
=
U
×
Σ
×
Document vectors
Term vectors
A (m × n)
U (m × m)
Σ (m × n)
V^T (n × n)

SVD分解降维

K=2：U: t*r→t*k;　S: r*r→k*k;　V^T: r*d→k*d;　**A?**

$$A_k = U_k * S_k * V_k^T$$

0.22	-0.11	3.34		0.20	0.61	0.46	0.54	0.28	0.00	0.02	0.02	0.08
0.20	-0.07		2.54	-0.06	0.17	-0.13	-0.23	0.11	0.19	0.44	0.62	0.53
0.24	0.04											
0.40	0.06											
0.64	-0.17				$k = 2$							
0.27	0.11											
0.27	0.11											
0.30	-0.14											
0.21	0.27											
0.01	0.49											
0.04	0.62											
0.03	0.45											

降维后文档间相似度（内积）

$$A^T A = (USV^T)^T (USV^T) = VS^T U^T USV^T = VS^2 V^T$$

$$M_k = (A^T \bullet A)_{(i,j)} = \sum_{x=1}^{k} V_{ix} \cdot V_{jx} \cdot S_{xx}^{\ 2}$$

	c1	c2	c3	c4	c5	m1	m2	M3	m4
c1	0.47	1.3	1.08	1.29	0.58	-0.03	-0.13	-0.2	-0.03
c2	1.3	4.34	2.99	3.42	2.03	0.34	0.62	0.82	1.13
c3	1.08	2.99	2.47	2.96	1.34	-0.06	-0.27	-0.42	-0.03
c4	1.29	3.42	2.96	3.59	1.52	-0.16	-0.53	-0.8	-0.3
c5	0.58	2.03	1.34	1.52	0.95	0.2	0.37	0.5	0.63
m1	-0.03	0.34	-0.06	-0.16	0.2	0.24	0.54	0.76	0.67
m2	-0.13	0.62	-0.27	-0.53	0.37	0.54	1.25	1.76	1.52
m3	-0.2	0.82	-0.42	-0.8	0.5	0.76	1.76	2.48	2.14
m4	-0.03	1.13	-0.03	-0.3	0.63	0.67	1.52	2.14	1.88

归一化（余弦）

$$M'_{(i,j)} = M_{(i,j)} \Big/ \sqrt{M_{(i,i)} \cdot M_{(j,j)}}$$

	c1	c2	c3	c4	c5	m1	m2	M3	m4
c1	**1.00**	0.91	0.99	0.99	0.87	-0.09	-0.16	-0.18	-0.03
c2	0.91	**1.00**	0.91	0.87	0.99	0.34	0.27	0.25	0.39
c3	0.99	0.91	**1.00**	0.99	0.88	-0.07	-0.15	-0.17	-0.02
c4	0.99	0.87	0.99	**1.00**	0.82	-0.17	-0.25	-0.27	-0.12
c5	0.87	0.99	0.88	0.82	**1.00**	0.41	0.34	0.33	0.47
m1	-0.09	0.34	-0.07	-0.17	0.41	**1.00**	0.99	0.99	0.99
m2	-0.16	0.27	-0.15	-0.25	0.34	0.99	**1.00**	0.99	0.99
m3	-0.18	0.25	-0.17	-0.27	0.33	0.99	0.99	**1.00**	0.99
m4	-0.03	0.39	-0.02	-0.12	0.47	0.99	0.99	0.99	**1.00**

查询：SVD空间变换

$$A = USV^T$$

- 变换函数：U_k^T

$$q'_k = U_k^T \bullet q \Leftrightarrow q'^T_k = q^T \bullet U_k$$

$$A'_k = U^T A = U^T USV_k^T = SV_k^T$$

$$q'^T_k = q^T \bullet U_k \qquad A'_k = SV_k^T$$

查询：SVD空间变换

$$A = USV^T$$

- 变换函数：U_k^T

$$q_k'^T = q^T \bullet U_k \qquad A_k' = SV_k^T$$

- 变换函数：$S^{-1}U_k^T$

$$q_k'^T = q^T \bullet U_k \bullet S^{-1} \qquad A_k'^T = V_k^T$$

- 变换函数：$S^{-1/2}U_k^T$

$$q_k'^T = q^T \bullet U_k \bullet S^{-1/2} \qquad A_k'^T = S^{-1/2}V_k^T$$

查询：SVD空间变换与降维

Query: Human-Computer Interaction

U_k

0.22	**-0.11**
0.20	**-0.07**
0.24	**0.04**
0.40	**0.06**
0.64	**-0.17**
0.27	**0.11**
0.27	**0.11**
0.30	**-0.14**
0.21	**0.27**
0.01	**0.49**
0.04	**0.62**
0.03	**0.45**

$\mathbf{q}^T$= 〈1 0 1 0 0 0 0 0 0 0 0 0〉

$$q_k'^T = q^T \bullet U_k$$

$$= < q^T \bullet u_1, q^T \bullet u_2, \cdots q^T \bullet u_k >$$

$\mathbf{q'}_k^T$= < 0.46 -0.07 >

SVD变换空间的相似度

$\mathbf{q'}_k^{\mathbf{T}} = < 0.46 \quad -0.07 >$

S_k

3.34	
	2.54

$$sim(Q, D_i) = (q_k^T \bullet (S \bullet V_k^T{}_{(i)})) \Big/ |q_k| \cdot |(S \bullet V_k^T{}_{(i)})|$$

V_k

c_1	c_2	c_3	c_4	c_5	m_1	m_2	m_3	m_4
0.20	0.61	0.46	0.54	0.28	0.00	0.02	0.02	0.08
-0.06	0.17	-0.13	-0.23	0.11	0.19	0.44	0.62	0.53

$sim(Q, D_i)$

c_1	c_2	c_3	c_4	c_5	m_1	m_2	m_3	m_4
0.99	**0.93**	**0.99**	**0.98**	**0.90**	**-0.01**	**-0.09**	**-0.11**	**0.04**

检索结果对比

Query: Human-Computer Interaction

q^T	(1 0 1 0 0 0 0 0 0 0 0 0)
0.82	*c*1– *human interface com*
0.35	*c*4– *system human system eps*
0.28	*c*2 – *survey user computer system response time*
0.00	*c*3 – *eps user interface system*
0.00	*c*5 – *user response time*
0.00	*m*1 – *trees*
0.00	*m*2 – *graph trees*
0.00	*m*3 – *graph minors trees*
0.00	*m*4– *graph minors survey*

$q'_k{}^T$	(0.46 −0.07)
0.99	**c3** – *eps user interface system*
0.99	*c*1 – *human interface computer*
0.98	*c*4 – *system human system eps*
0.93	*c*2 – *survey user computer system response time*
0.90	**c5** – *user response time*
0.04	*m4* – *graph minors survey*
-0.01	*m1* – *trees*
-0.09	*m2* – *graph trees*
-0.11	*m*3 – *graph minors trees*

查询质量

- 查询质量的变化?

查询：SVD空间变换（2）

Query: Human-Computer Interaction

$\mathbf{q}^T = \langle 1\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0 \rangle$

$$q_k'^T = q^T \bullet U_k \bullet S^{-1}$$

$$= < q^T \bullet u_1 \cdot s_{11}^{-1}, q^T \bullet u_2 \cdot s_{22}^{-1}, \cdots q^T \bullet u_k \cdot s_{kk}^{-1} >$$

$\mathbf{q'}_k^T = < 0.1377 \quad -0.0276 >$

U_k

0.22	**-0.11**
0.20	**-0.07**
0.24	**0.04**
0.40	**0.06**
0.64	**-0.17**
0.27	**0.11**
0.27	**0.11**
0.30	**-0.14**
0.21	**0.27**
0.01	**0.49**
0.04	**0.62**
0.03	**0.45**

S_k

3.34	
	2.54

SVD变换空间的相似度（2）

$\mathbf{q'}_k^{\mathbf{T}} = < 0.1377 \quad -0.0276 >$

$$sim(Q, D_i) = (q_k^T \bullet V_{k(i)}) \Big/ |q_k| \cdot |V_{k(i)}|$$

V_k

c_1	c_2	c_3	c_4	c_5	m_1	m_2	m_3	m_4
0.20	0.61	0.46	0.54	0.28	0.00	0.02	0.02	0.08
-0.06	0.17	-0.13	-0.23	0.11	0.19	0.44	0.62	0.53

$sim(Q, D_i)$

c_1	c_2	c_3	c_4	c_5	m_1	m_2	m_3	m_4
0.99	**0.89**	**0.99**	**0.96**	**0.84**	**-0.20**	**-0.15**	**-0.16**	**-0.05**

检索结果对比

$$q_k'^{T} = q^{T} \bullet U_k$$

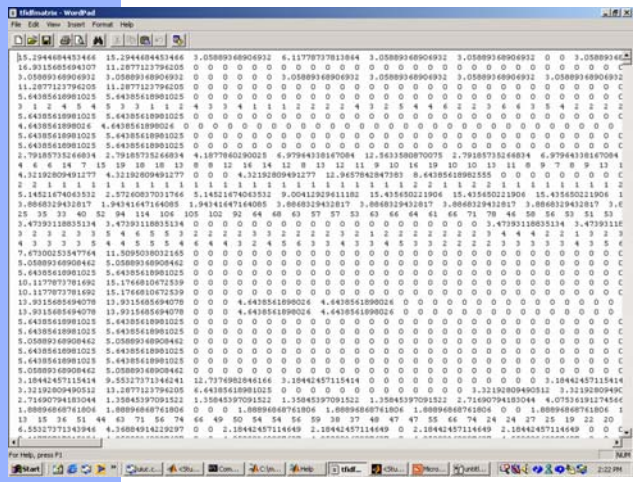
$q'_k{}^{T}$	(0.46 −0.07)
0.99	*c*3 – *eps user interface system*
0.99	*c*1 – *human interface computer*
0.98	*c*4 – *system human system eps*
0.93	*c*2 – *survey user computer system response time*
0.90	*c*5 – *user response time*
0.04	*m4* – *graph minors survey*
-0.01	*m1* – *trees*
-0.09	*m2* – *graph trees*
-0.11	*m3* – *graph minors trees*

$$q_k'^{T} = q^{T} \bullet U_k \bullet S^{-1}$$

$q'_k{}^{T}$	(0.1377 -0.0276)
0.99	*c*3 – *eps user interface system*
0.99	*c*1 – *human interface computer*
0.96	*c*4 – *system human system eps*
0.89	*c*2 – *survey user computer system response time*
0.84	*c*5 – *user response time*
-0.05	*m4* – *graph minors survey*
-0.15	*m2* –*graph trees*
-0.16	*m3* – *graph minors trees*
-0.20	*m1* – *trees*

SVD变换后的词空间与文档空间

Weighted Frequency Matrix



DOCUMENTS:

'CM031.txt'

'CM046.txt'

'CM001.txt'

'CM029.txt'

'CM040.txt'

K>> return

TERMS:

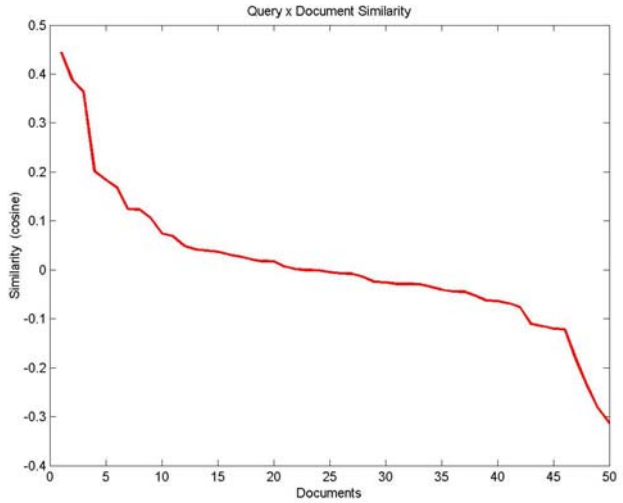
'joint'

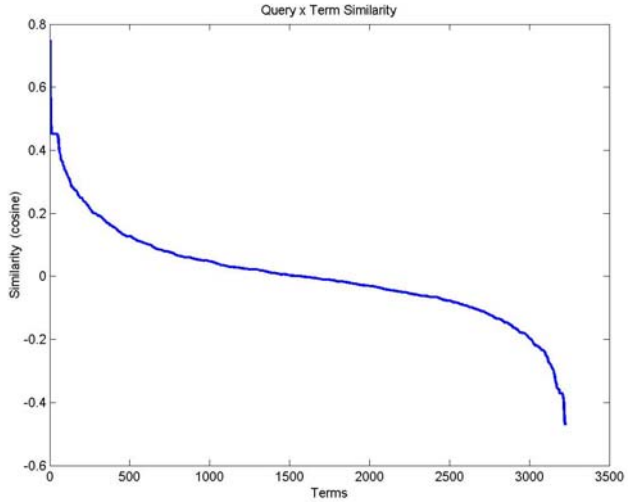
'insulation'

'roofing'

'expansion'

'saw'





Query Terms:
- **Insulation**
- **Joint**

词间关系发生变化

Any Question?