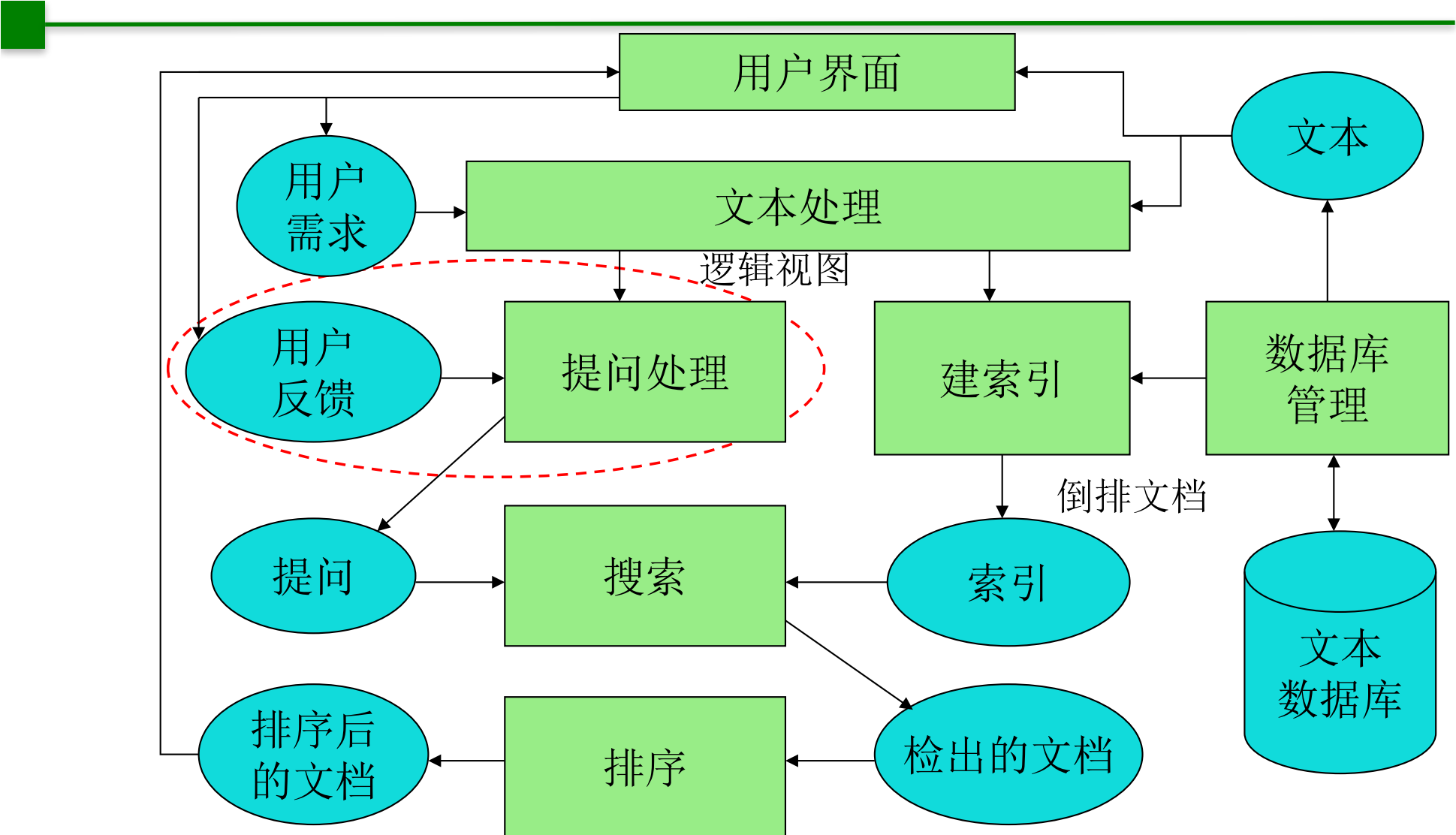


信息检索 Information Retrieval



第五章 相关反馈与查询扩展

| Age Group | Percentage |
|-----------|------------|
| 18-24 | 18% |
| 25-34 | 22% |
| 35-44 | 15% |
| 45-54 | 12% |
| 55-64 | 10% |
| 65-74 | 8% |
| 75-84 | 5% |
| 85+ | 3% |



引言

- 为了实现对信息进行检索，大多数用户发现很难给出一个好的查询词
- 然而，大多数用户通常需要重新编写查询，以获得他们感兴趣的结果
 - 因此，用户给出的第一个查询应被视为检索相关信息的初始尝试
 - 对最初检索到的文档可以进行相关性分析，并用于改进初始查询

引言

- 对用户查询的修改通常是指

- 相关反馈

- 针对一个给定的查询，用户提供哪些文档时相关的、哪些文档是不相关的信息

- 显式反馈

- 用户直接提供与查询重构有关的信息

- 隐式反馈

- 与查询重构有关的信息由系统隐式给出

- 查询扩展

- 利用与查询相关的信息来扩展查询词

主要内容

- 相关反馈
- 查询扩展

相关反馈

● 相关反馈的框架

- 给定一个查询 q ， D_r 是与 q 相关的文档集合
- 在相关反馈过程中，使用 D_r 来的对原始查询 q 修改后的查询 q_m
- 然而，哪些文档是与当前查询是相关的？需要用户的直接介入
 - 大多数的用户都不希望提供这样的信息，尤其在互联网环境中

相关反馈

● 相关反馈的框架

- 除了用户的直接介入，我们还可以通过其他方式来知道哪些文档是相关的
 - 用户点击了哪些文档
 - ▣ 用户点击了一篇文档，意味着对于当前的查询，该文档使用户感兴趣的（但不一定是相关文档）
 - 返回结果中排序靠前的文档中包含的词项
- 最终的目的就是希望得到的结果质量更高

相关反馈

● 相关反馈的框架

■ 一个反馈循环（feedback cycle）包括两个基本步骤

➤ 确定与原始查询 q 相关的反馈信息

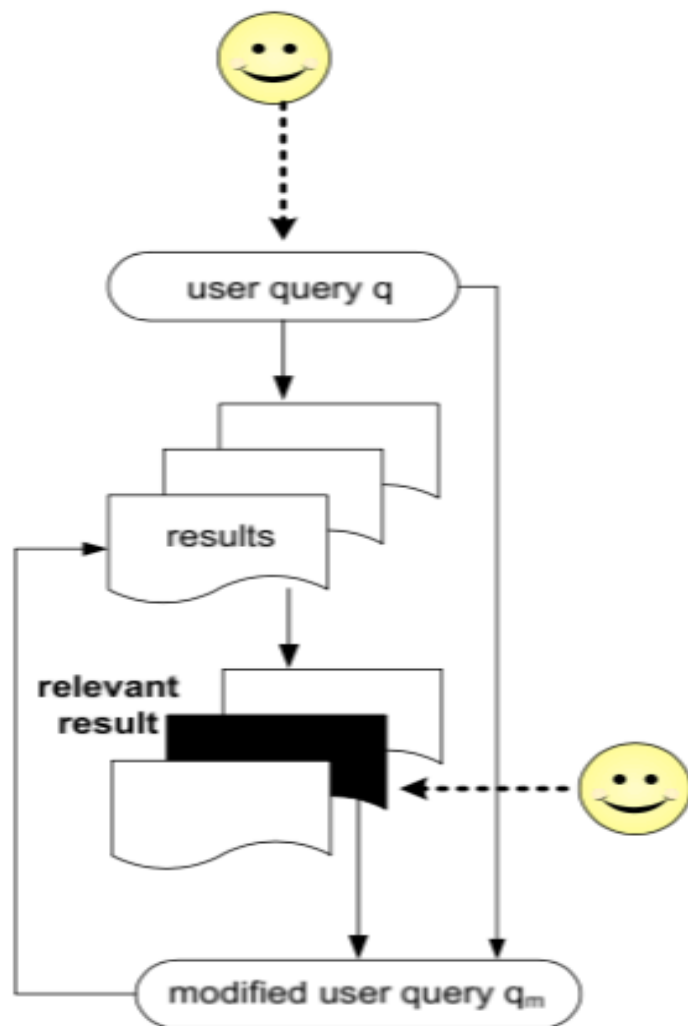
- 用户显式地直接给出反馈信息

- 从系统返回结果或者从词库（thesaurus）中获得反馈信息

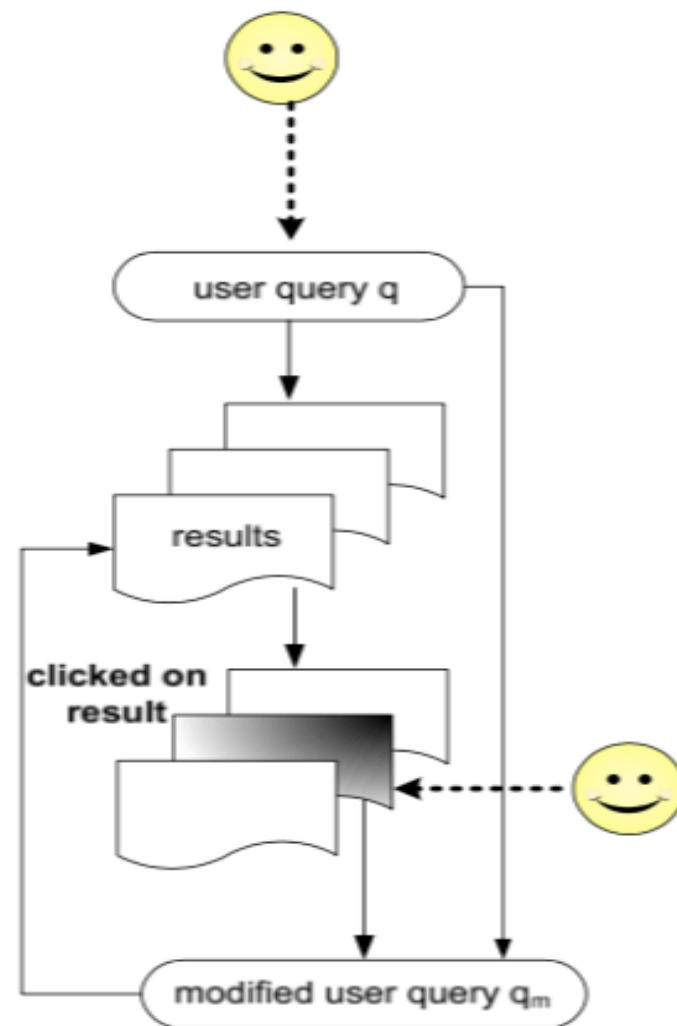
➤ 确定如何转换查询 q 以有效地利用这些信息

相关反馈

显式反馈



(a) relevance feedback



(b) click feedback

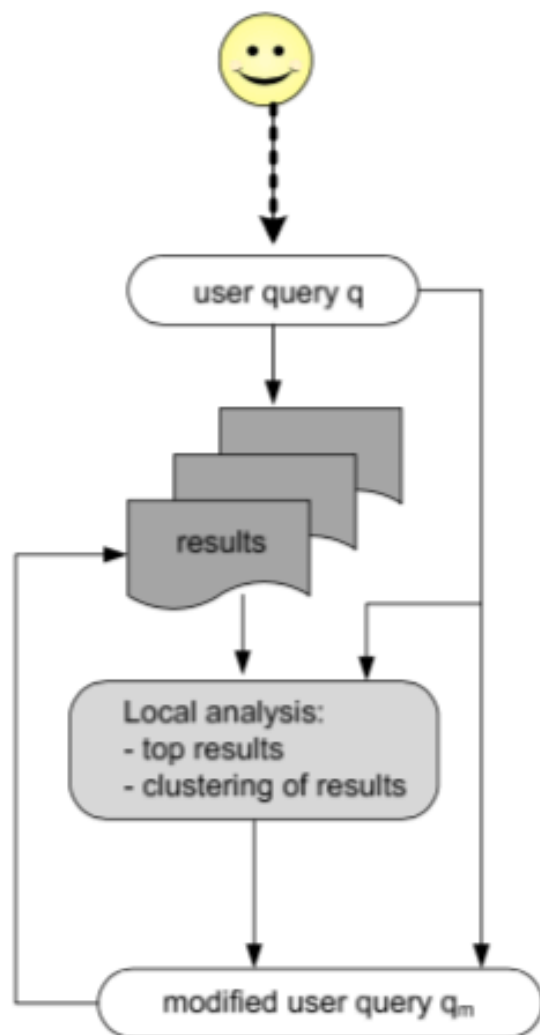
相关反馈

● 相关反馈的框架

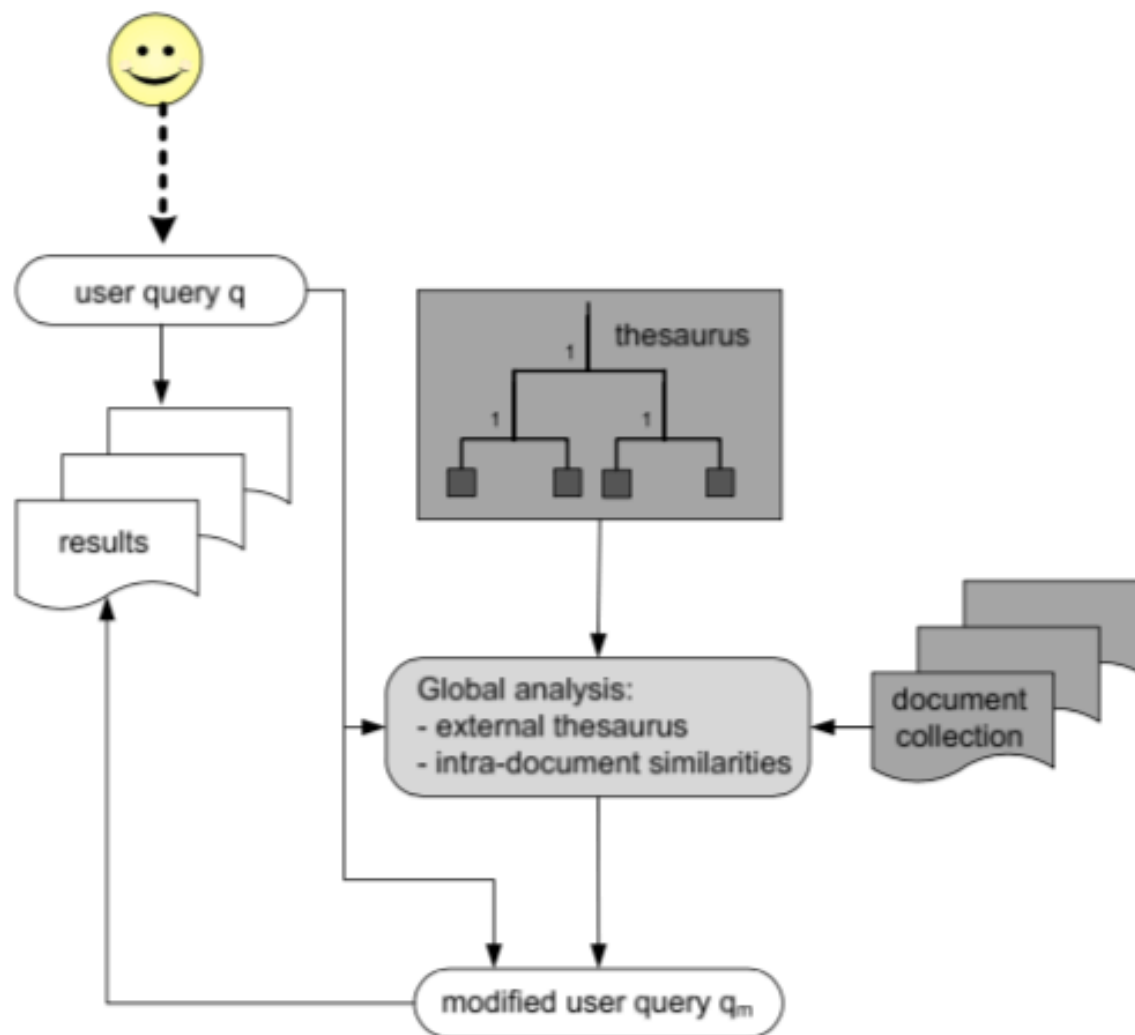
- 在隐式反馈中，系统隐式地给出反馈信息
- 获得隐式反馈信息有两种基本方法
 - 从结果集中排名靠前的文档中获取反馈信息
 - 从外部源（如同义词库）获取反馈信息

相关反馈

隐式反馈



(a) local analysis



(b) global analysis

Explicit Relevance Feedback

显式相关反馈

显式相关反馈

- 在一个典型的相关反馈周期中
 - 用户会得到一个检索到的文档列表
 - 然后，用户检查它们并标记哪些文档是相关的
 - 实际上，只需要检查排名前10（或20）的文档
- 主要思想
 - 从已确定为相关的文件中选择重要词项
 - 增强这些词项在新构造的查询中的重要性
- 显示反馈的预期效果
 - 新构造的查询将更加偏向于向相关文档，而不是非相关文档

显式相关反馈

● Rocchio Method

- 对于给定的查询，确定为相关的文档彼此之间有相似之处
- 此外，非相关文档中的词项权重向量与相关文档不同
- Rocchio方法的基本思想是重新构造查询，使其得到：
 - 在向量空间中更加邻近相关文档、远离非相关文档

显式相关反馈

● Rocchio Method

■ 质心

➤ 质心指的是一系列点的中心

$$\vec{\mu}(D) = \frac{1}{|D|} \sum_{d \in D} \vec{v}(d)$$

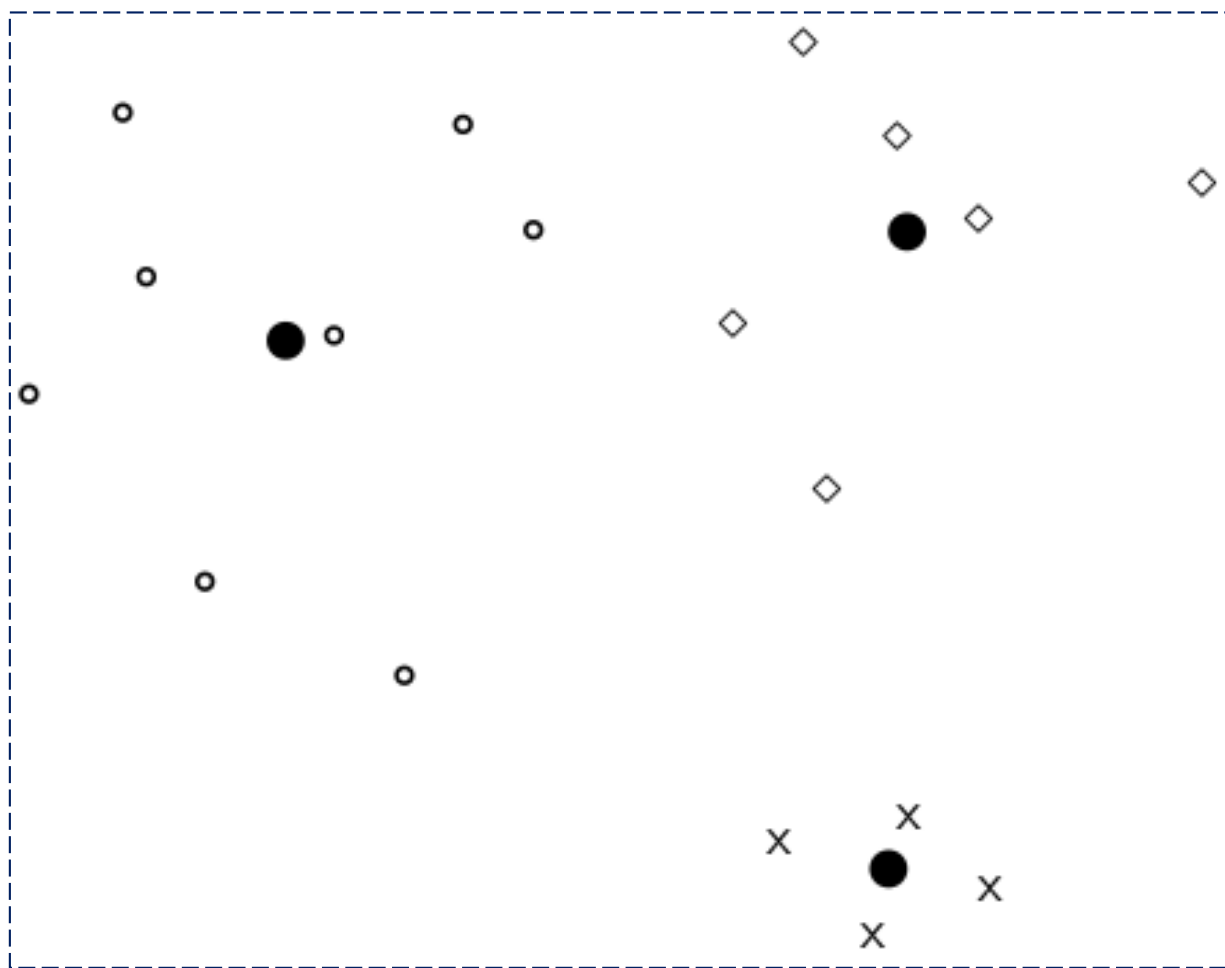
□ D 是一个文档集合

□ $\vec{v}(d) = \vec{d}$ 文档 d 的向量表示

显式相关反馈

- Rocchio Method

- 质心的例子



显式相关反馈

● Rocchio Method

- Rocchio算法是向量空间模型中相关反馈的实现方式
- Rocchio算法使如下的方法得到新的查询

$$\vec{q}_{opt} = \underset{\vec{q}}{argmax} [sim(\vec{q}, \mu(D_r)) - sim(\vec{q}, \mu(D_{nr}))]$$

➤ D_r : 相关文档集; D_{nr} : 不相关文档集

- \vec{q}_{opt} 是使得相关文档和不相关文档区分度最大的向量

显式相关反馈

● Rocchio Method

- 最优查询向量为:

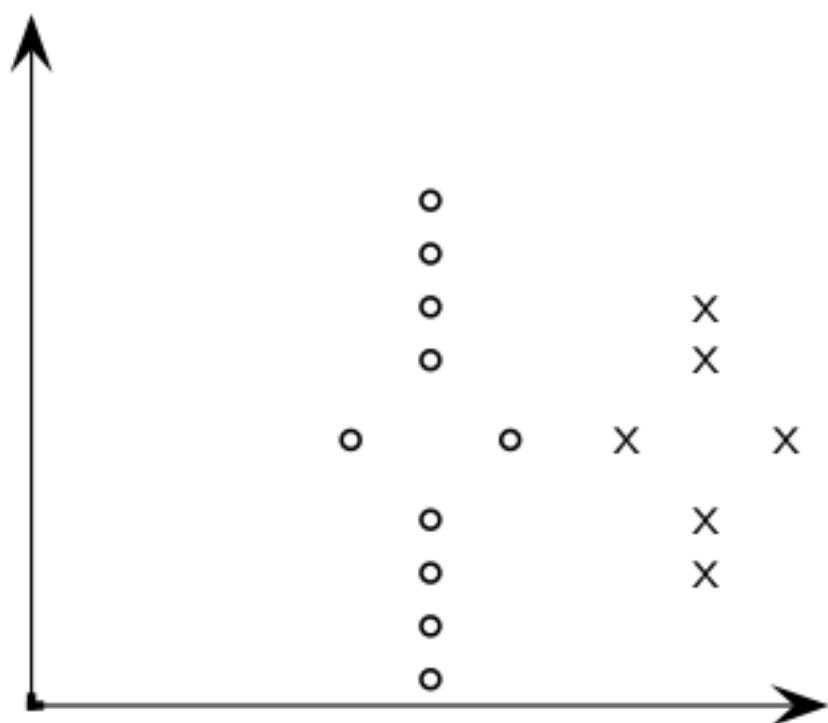
$$\begin{aligned}\vec{q}_{opt} &= \mu(D_r) + [\mu(D_r) - \mu(D_{nr})] \\ &= \frac{1}{|D_r|} \sum_{\vec{d}_j \in D_r} \vec{d}_j + \left[\frac{1}{|D_r|} \sum_{\vec{d}_j \in D_r} \vec{d}_j - \frac{1}{|D_{nr}|} \sum_{\vec{d}_j \in D_{nr}} \vec{d}_j \right]\end{aligned}$$

- 即将相关文档的质心移动一个量，该量为相关文档质心和不相关文档的差异量

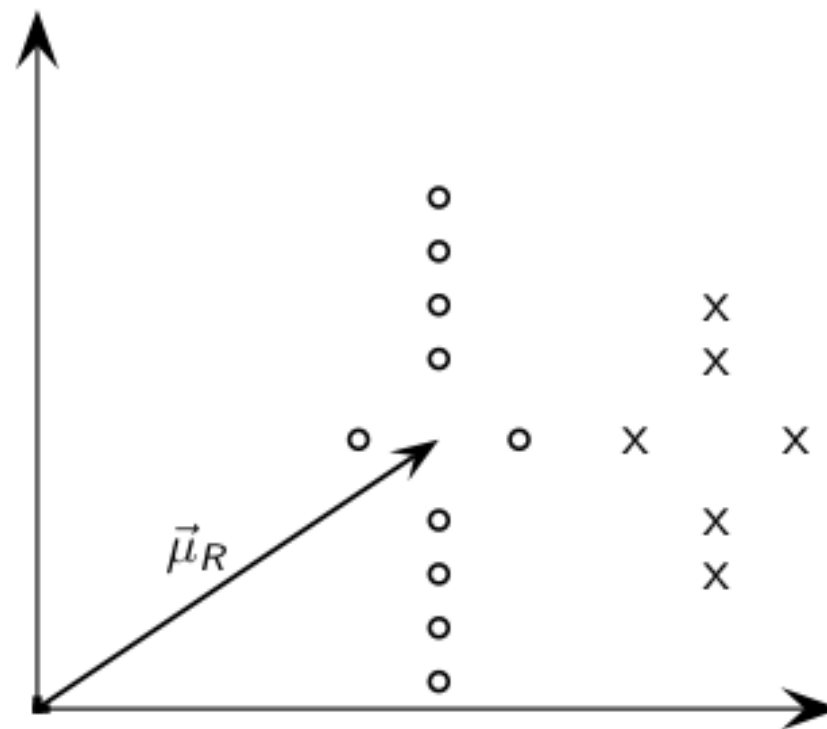
显式相关反馈

● Rocchio Method

■ \vec{q}_{opt} 计算示例



○：相关文档；×：不相关文档

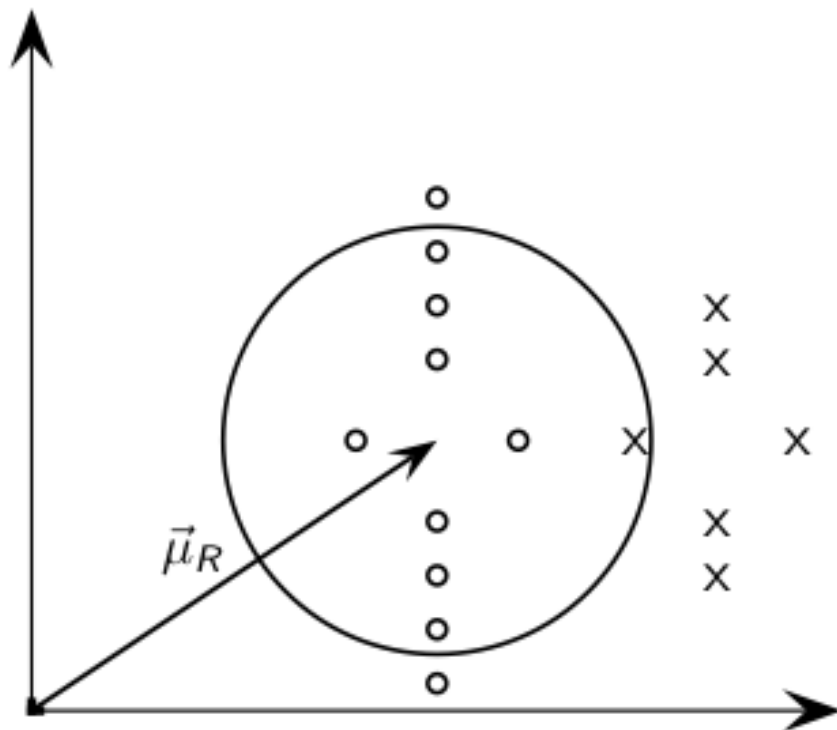


$\vec{\mu}_R$ ：相关文档的质心

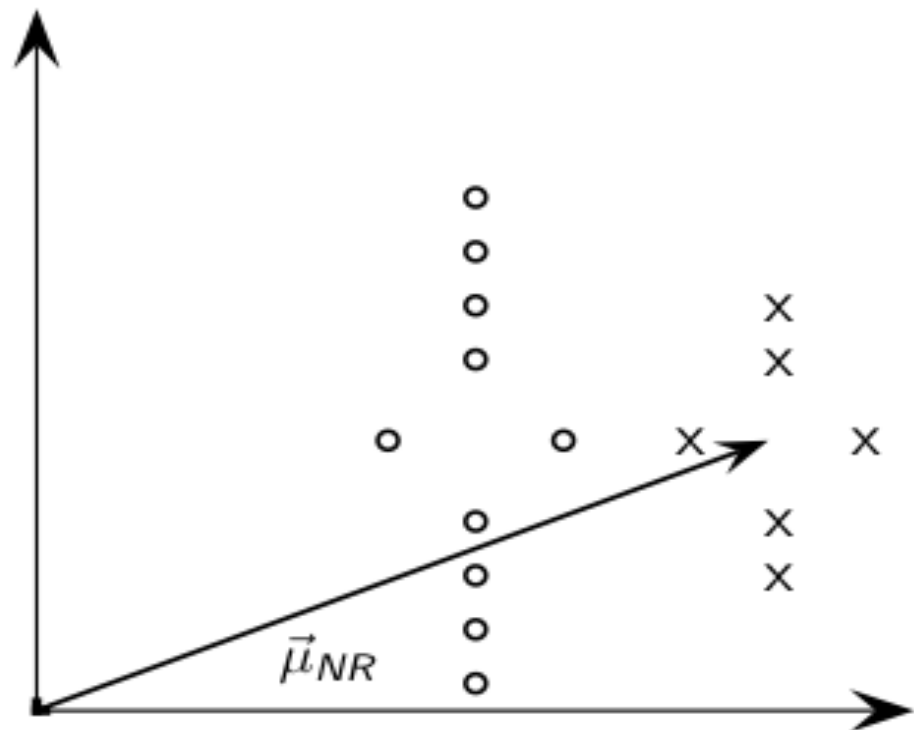
显式相关反馈

● Rocchio Method

■ \vec{q}_{opt} 计算示例



$\vec{\mu}_R$ 不能将相关文档和不相关文档分开

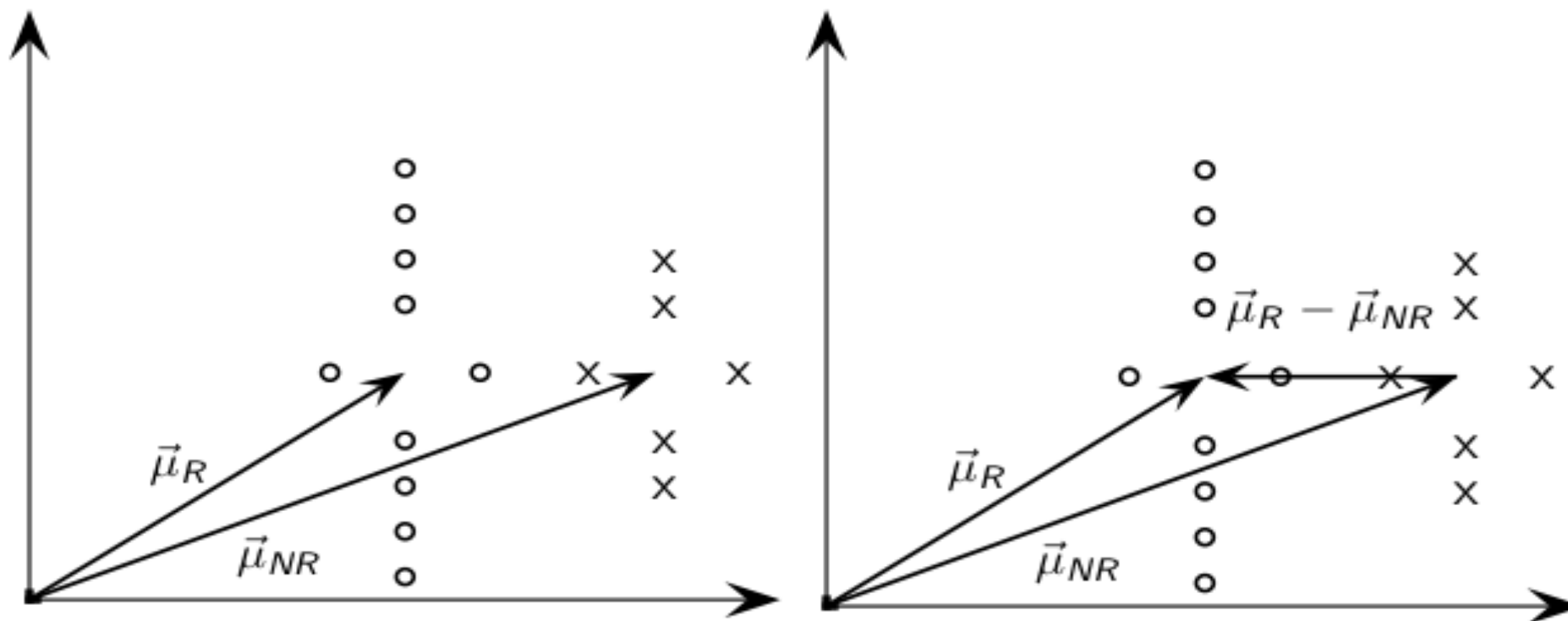


$\vec{\mu}_{NR}$: 不相关文档的质心

显式相关反馈

● Rocchio Method

■ \vec{q}_{opt} 计算示例



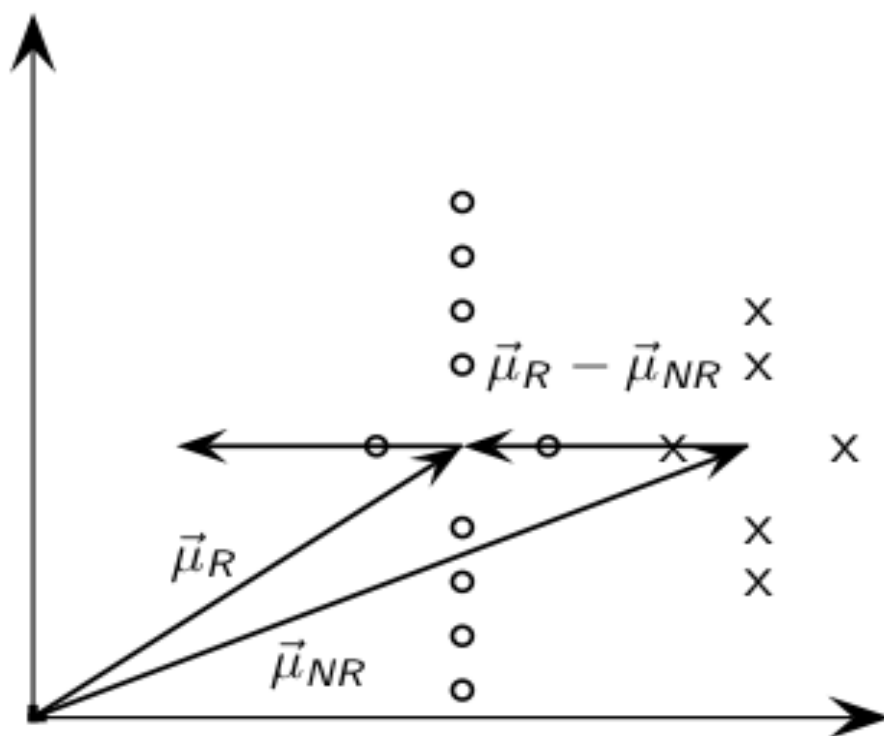
$\vec{\mu}_R$: 相关文档质心向量;
 $\vec{\mu}_{NR}$ 不相关文档质心向量;

差异向量: $\vec{\mu}_R - \vec{\mu}_{NR}$

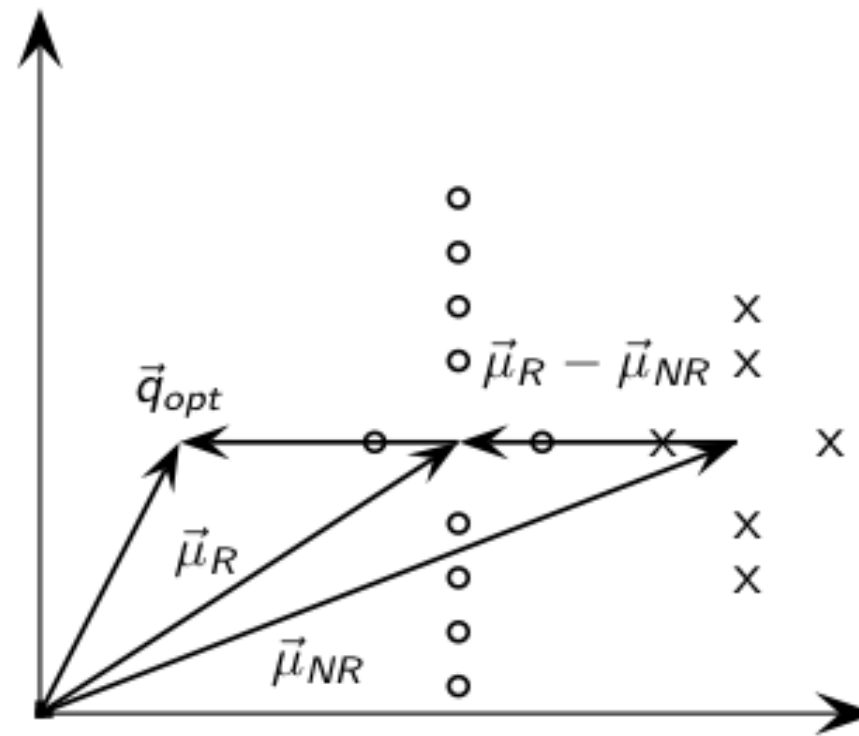
显式相关反馈

● Rocchio Method

■ \vec{q}_{opt} 计算示例



$\vec{\mu}_R$ 加上差异向量

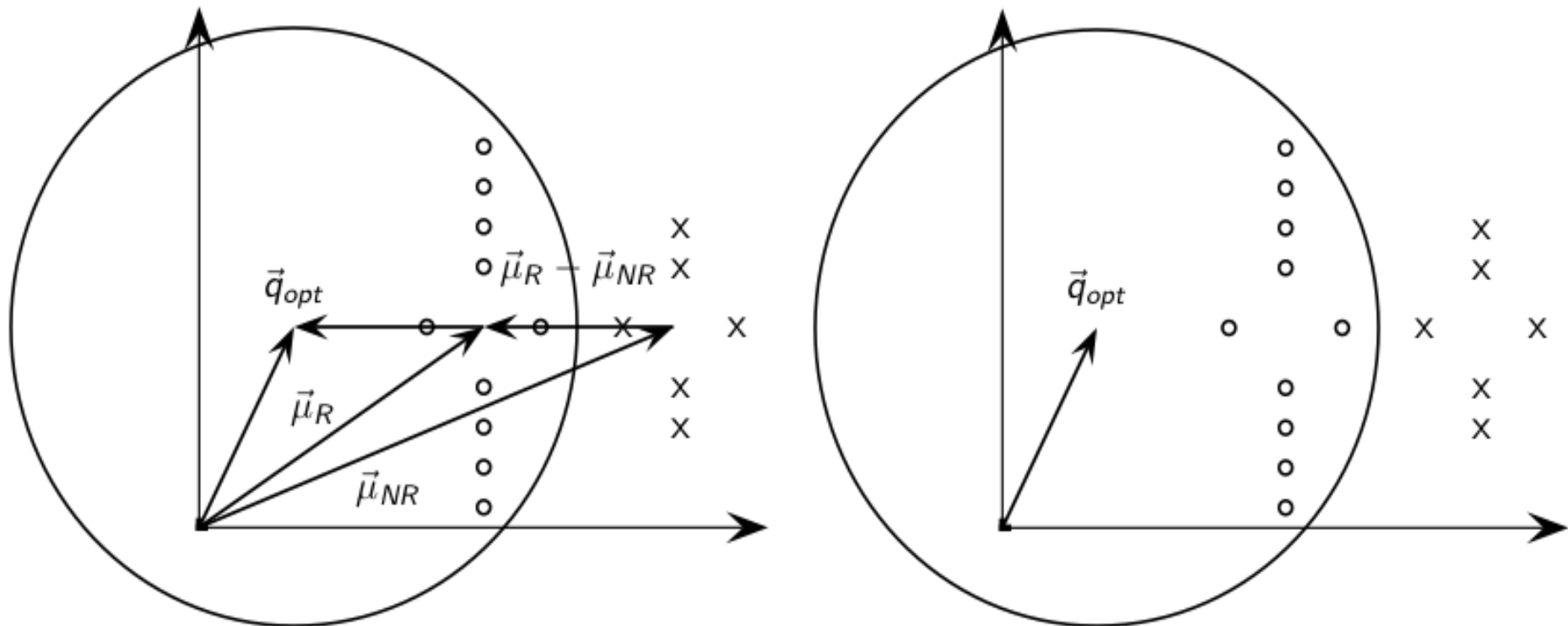


得到 \vec{q}_{opt}

显式相关反馈

● Rocchio Method

■ \vec{q}_{opt} 计算示例



\vec{q}_{opt} 能够将相关文档和不相关文档完美分开

显式相关反馈

● Rocchio Method

- 实际使用中的公式（1971年，SMART系统使用）

$$\begin{aligned}\vec{q}_m &= \alpha \vec{q}_0 + \beta \mu(D_r) - \gamma \mu(D_{nr}) \\ &= \alpha \vec{q}_0 + \beta \frac{1}{|D_r|} \sum_{\vec{d}_j \in D_r} \vec{d}_j - \gamma \frac{1}{|D_{nr}|} \sum_{\vec{d}_j \in D_{nr}} \vec{d}_j\end{aligned}$$

- \vec{q}_m : 修改后的查询; \vec{q}_0 : 原始查询
- D_r 、 D_{nr} : 已知的相关文档和不相关文档集合
- α 、 β 、 γ : 权重
 - α 、 β 、 γ 设置的折中: 如果判定的文档数目很多, 那么可以考虑 β 、 γ 设置得大一些
 - 计算后如果权重为负, 则将权重都设置为0

显式相关反馈

● Rocchio Method

■ 正反馈 (Positive) Vs. 负反馈 (Negative)

- 正反馈价值往往大于负反馈
- 比如, 可以通过设置 $\beta = 0.75$, $\gamma = 0.25$ 来给正反馈更大的权重
- 很多系统甚至只允许正反馈, 即 $\gamma=0$

显式相关反馈

● 相关反馈的评价

- 选选择“信息检索的评价”中的某个评价指标，比如 $P@10$
 - 计算原始查询 q_0 检索结果的 $P@10$ 指标
 - 计算修改后查询 q_1 检索结果的 $P@10$ 指标
- 大部分情况下 q_1 的检索结果精度会显著高于 q_0
- 上述评价过程是否公平？

显式相关反馈

- 相关反馈存在的问题
 - 相关反馈开销很大
 - 相关反馈生成的新查询往往很长
 - 长查询的处理开销很大
 - 用户不愿意提供显式的相关反馈
 - *Excite*搜索引擎曾经提供完整的相关反馈功能，但是后来废弃了这一功能

显式相关反馈

- 通过用户点击获得显式反馈
 - 鼠标键盘动作
 - 点击链接
 - 加入收藏夹
 - 拷贝粘贴
 - 停留
 - 翻页
 -

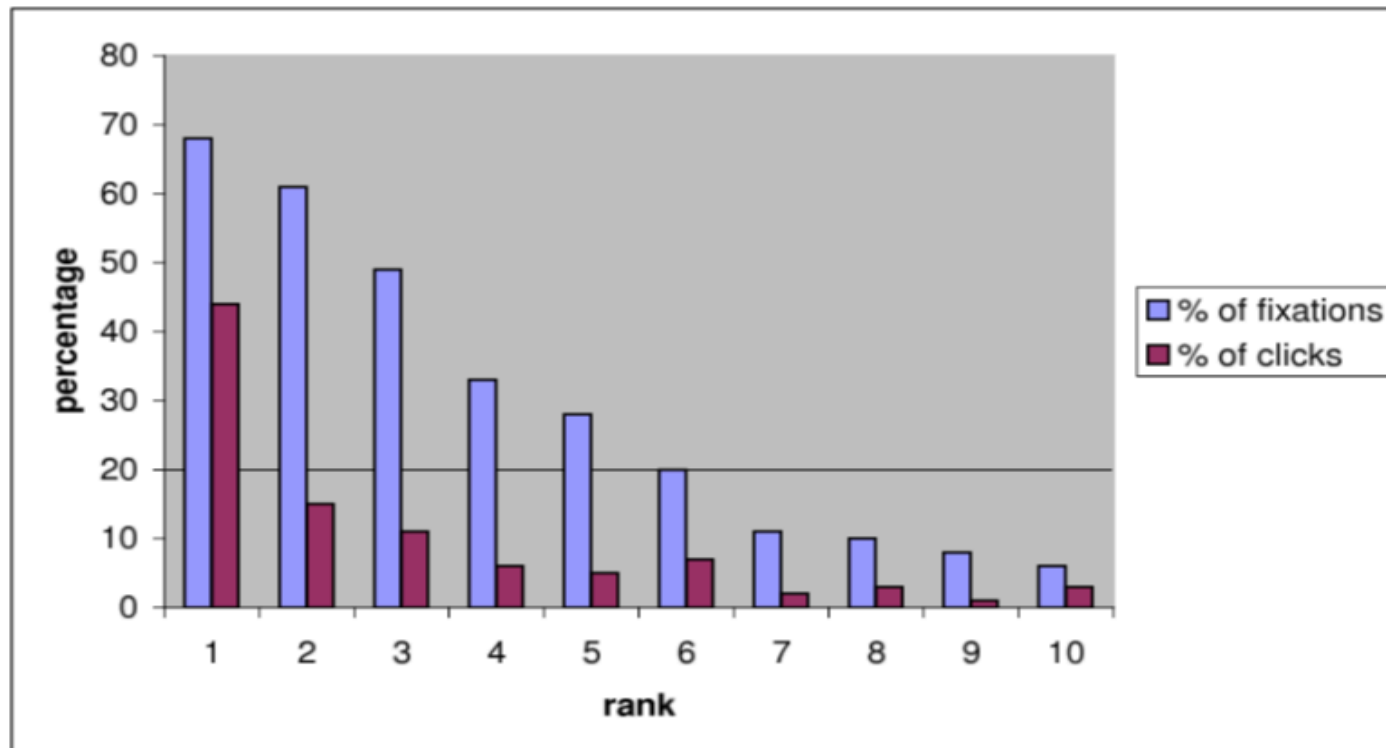
显式相关反馈

- 通过用户点击获得显式反馈
 - 网络搜索引擎用户不仅检查系统返回的答案，也会点击感兴趣的链接
 - 点击行为反映了用户对于某个结果的偏好
 - 可以在不干扰用户的情况下大量收集这些信息
 - 问题是，它们是否能够反映某个结果与查询的相关性
 - 在通过点击获得显示反馈信息的任务中，答案是肯定的

显式相关反馈

- 通过用户点击获得显式反馈
 - 用户行为分析

10个任务中，用户浏览和点击结果的比率

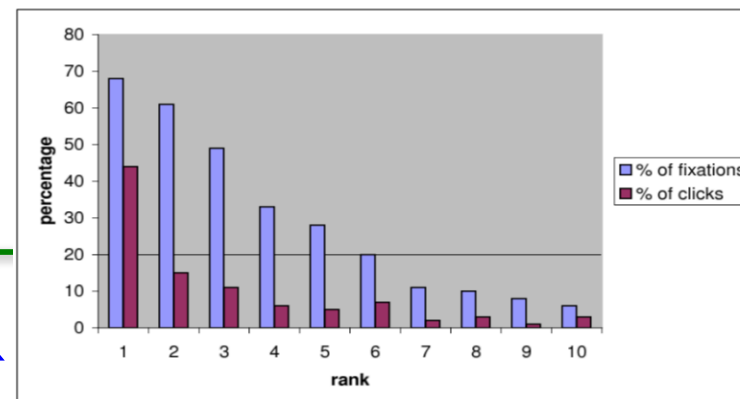


显式相关反馈

- 通过用户点击获得显式反馈

- 用户行为分析

- 用户审视前两个结果的比例基本相等，但是点击第一个结果的比例是第二个结果的3倍左右
 - 这些信息表明用户对搜索引擎偏见
 - 也就是说，用户倾向于相信搜索引擎推荐的靠前的结果是相关的

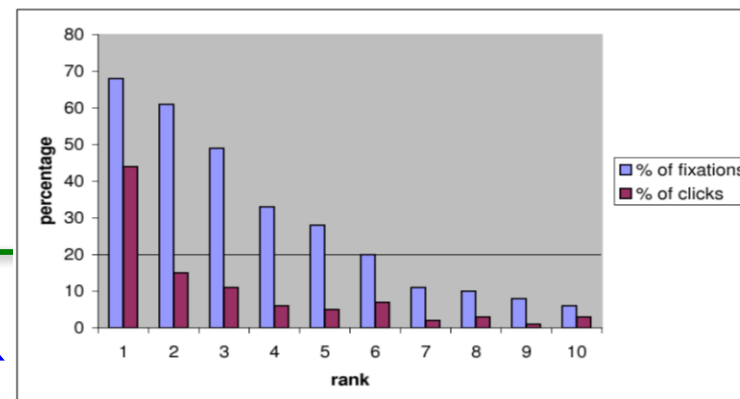


显式相关反馈

● 通过用户点击获得显式反馈

■ 用户行为分析

- 通过给出两个不同的结果集，可以更好地理解这一点
 - 一个是搜索引擎返回的正常排名
 - 一个是修改后的排名，其中前2个结果交换了它们的位置
- 分析表明，用户的偏好仍然是排名靠前的结果
- 也就是说，结果的位置对于用户是否点击它有很大的影响



显式相关反馈

- 通过用户点击获得显式反馈
 - 将点击行为看做是用户偏好的度量
 - 很明显，将用户的点击行为解释为结果与查询的相关性是不合理的
 - 将点击理解为用户偏好的度量标准更合适
 - 这种类型的偏好关系考虑到
 - 用户点击的结果
 - 已检查但未点击的结果

显式相关反馈

- 通过用户点击获得显式反馈

- 同一查询中的点击行为

- 给出如下定义

- $R(q_i, d_j)$ 给定一个排序函数, r_k 是结果中排序为 k 的结果

- $\sqrt{r_k}$ 表示用户点击了第 k 个结果

- 定义偏好函数 $r_k > r_{k-n}$, $0 < k-n < k$

- 根据用户的点击行为, 相比于第 $(k-n)$ 个结果, 用户更偏好第 k 个结果

显式相关反馈

● 通过用户点击获得显式反馈

■ 同一查询中的点击行为

➤ 示例: $r_1, r_2, \sqrt{r_3}, r_4, \sqrt{r_5}, r_6, r_7, r_8, r_9, \sqrt{r_{10}}$

□ 我们并不能根据点击的结果就确定 r_3, r_5, r_{10} 与查询是相关的

➤ 两个用于确定偏好关系的策略

□ Skip-above: 如果 $\sqrt{r_k}$, 那么 $r_k > r_{k-n}$ (对于所有没有点击的 r_{k-n}) ($r_3 > r_2, r_3 > r_1$)

□ Skip-Previous: 如果 $\sqrt{r_k}$, 那么 $r_k > r_{k-1}$ (如果 r_{k-1} 没有被点击) ($r_3 > r_2$)

□ Skip-above能够生成更多的偏好关系

显式相关反馈

● 通过用户点击获得显式反馈

■ 同一查询中的点击行为

- 经验结果表明，在大约80%的实例中，用户点击与结果相关性的判断一致
 - 上面的Skip-above和Skip-Previous策略都会产生偏好关系
 - 交换第一个和第二个结果，对于这两个策略，点击仍然会反映出偏好关系
 - 如果颠倒前10个结果的顺序，点击仍然会反映出两种策略的偏好关系
- 因此，用户的点击行为不仅可以很好地反映个人的偏好，还可以用作对给定查询结果的相关性的判定

显式相关反馈

- 通过用户点击获得显式反馈
 - 查询链（query chain）中的点击行为
 - 在实际应用中，用户在搜索同一任务的答案时会提出多个查询
 - 可以在实时查询流中标识与同一任务关联的查询集
 - ▣ 这个集合构成了所谓的查询链
 - 分析查询链的目的是产生新的偏好关系

显式相关反馈

- 通过用户点击获得显式反馈
 - 查询链（query chain）中的点击行为

- 查询链中的两个结果集示例

$r_1, r_2, r_3, r_4, r_5, r_6, r_7, r_8, r_9, r_{10}$

$s_1, \sqrt{s_2}, s_3, s_4, \sqrt{s_5}, s_6, s_7, s_8, s_9, s_{10}$

□ r_j 是第一个结果集合

□ s_j 是第二个结果集合

- 获取偏好关系的两个策略

□ Top-One-No-Click-Earlier

– if $\exists sk | \forall sk$ then s_j

□ Top--Two-No-Click-Earlier

– if $\exists sk | \forall sk$ then $s_j > r_1$ and $s_j > r_2$ for $j \leq 10$

$s_1 > r_1, s_2 > r_1, s_3 > r_1, s_4 > r_1, s_5 > r_1, \dots$

$s_1 > r_1, s_2 > r_1, s_3 > r_1, s_4 > r_1, s_5 > r_1, \dots$

$s_1 > r_2, s_2 > r_2, s_3 > r_2, s_4 > r_2, s_5 > r_2, \dots$

显式相关反馈

● 通过用户点击获得显式反馈

■ 查询链（query chain）中的点击行为

- 我们注意到，第二种策略产生的偏好关系比第一种策略多两倍
- 这些偏好关系必须与人工给出的相关性判断结果进行比较
- 得出以下结论
 - 在大约80%的实例中，这两种策略产生的偏好关系与相关性判断一致
 - 即使交换第一和第二个结果，也会得到同样的结论
 - 即使颠倒结果的顺序，也会得到同样的结论

implicit Relevance Feedback

隐式相关反馈

隐式相关反馈

- 自动局部分析 (local Analysis) 的隐式相关反馈
 - 给定查询 q ，从检索到的文档中获取的反馈信息进行局部分析
 - 类似于一个相关反馈循环周期，在没有用户参与的情况下完成
 - 两种局部策略
 - 局部聚类
 - 局部上下文分析

隐式相关反馈

● 自动局部分析 (local Analysis) 的隐式相关反馈

■ 局部聚类

- 采用聚类技术进行查询扩展是信息检索的基本方法
- 量化词项之间相关性，然后使用相关的词项进行查询扩展
- 词项之间的相关性可以通过使用全局结构（如关联矩阵）来量化

隐式相关反馈

● 自动局部分析 (local Analysis) 的隐式相关反馈

■ 局部聚类

➤ 对于给定的一个查询 q

□ D_l : 局部文档集合 (针对查询 q 检索回来的文档集合)

□ N_l : D_l 中的文档个数

□ V_l : 局部词表 (D_l 中的不同的单词集合)

□ $f_{i,j}$: 词项 k_i 在文档 $D_j \in D_l$ 中出现的次数

□ $M_l = [m_{ij}]$: 词项-文档矩阵 (V_l 行, N_l 列)

□ $m_{ij} = f_{i,j}$: 矩阵 M_l 中的元素

□ M_l^T : 矩阵 M_l 的转置矩阵

➤ 局部词项-词项关联矩阵

$$C_l = M_l M_l^T$$

隐式相关反馈

$$C_l = M l M_l^T$$

● 自动局部分析 (local Analysis) 的隐式相关反馈

■ 局部聚类

- 矩阵中的元素 $c_{u,v} \in C_l$ 表示词项 k_u 和 k_v 之间的关联程度
- 两个词项之间的关系是基于它们在文档集合中的共同出现
- 两个词项同时出现的文档数越多，相关性就越强
- 同一簇中的词项可用于查询扩展
- 我们在这里考虑三种类型的簇
 - 关联簇 (Association Clusters)
 - 度量簇 (Metric Clusters)
 - 标量簇 (Scalar Clusters)

隐式相关反馈

● 自动局部分析 (local Analysis) 的隐式相关反馈

■ 局部聚类—关联簇

- 关联簇通过局部关联矩阵 C_l 的计算得到
- 词项 k_u 和 k_v 之间的关联因子

$$c_{u,v} = \sum_{d_j \in D_l} f_{u,j} \times f_{v,j}$$

- 在这种情况下，相关矩阵被称为局部关联矩阵
- 基本思想
 - 文档中经常共同出现的词项具有同义关联

隐式相关反馈

- 自动局部分析 (local Analysis) 的隐式相关反馈
 - 局部聚类—度量簇 (Metric Clusters)
 - 关联簇中只考虑了两个词项是否共现，但没有考虑词项在文档中出现的位置
 - 同一句话中出现的两个术语往往相关性更强
 - 度量簇重新定义了相关系数 $c_{u,v}$ ，作为两个词项在文档中距离的函数

隐式相关反馈

● 自动局部分析 (local Analysis) 的隐式相关反馈

■ 局部聚类—度量簇 (Metric Clusters)

- 令函数 $k_u(n, j)$ 返回词项 k_u 在文档 d_j 中出现的第 n 个位置
- 令函数 $r(k_u(n, j), k_v(m, j))$ 计算词项之间的距离
 - 词项 k_u 在文档 d_j 中的第 n 次出现
 - 词项 k_v 在文档 d_j 中的第 m 次出现
- 度量簇关联矩阵

$$c_{u, v} = \sum_{d_j \in D_I} \sum_n \sum_m \frac{1}{r(k_u(n, j), k_v(m, j))}$$

隐式相关反馈

● 自动局部分析 (local Analysis) 的隐式相关反馈

■ 局部聚类—标量簇 (Scalar Clusters)

- 通过比较两个词项的邻域 (neighborhoods)，也可以得到两个词项之间的相关性
- 两个具有相似邻域关系的词项有可能具有同义关系
 - 这种关系是间接的，或者是由邻域得到的
 - 通过一个标量来量化两个词项邻域的关系
 - 两个向量之间夹角的余弦是一种常用方法来度量的标量相似性

隐式相关反馈

- 自动局部分析 (local Analysis) 的隐式相关反馈

- 局部聚类—标量簇 (Scalar Clusters)

- $\vec{s}_u = (s_{u,x1}, s_{u,x2}, \dots, s_{u,xn})$ 是词项 k_u 的邻域关联值向量
 - $\vec{s}_v = (s_{v,y1}, s_{v,y2}, \dots, s_{v,yn})$ 是词项 k_v 的邻域关联值向量
 - 局部标量矩阵

$$c_{u,v} = \frac{\vec{s}_u \cdot \vec{s}_v}{|\vec{s}_u| \times |\vec{s}_v|}$$

隐式相关反馈

● 自动局部分析 (local Analysis) 的隐式相关反馈

■ 局部上下文分析

- 局部聚类技术基于针对某一查询检索到的文档集合
- 局部上下文分析是一种将全局分析与局部分析相结合的方法
 - 它基于名词词组的使用
 - 即文本中的单个名词、两个名词或三个相邻名词
 - 从排名靠前的文档中选择的名词词组，并把它作为文档的概念
 - 然而，这里不用文献，而是用段落来确定同时出现的信息
 - 段落是固定长度的文本

隐式相关反馈

● 自动局部分析 (local Analysis) 的隐式相关反馈

■ 局部上下文分析

➤ 局部上下文分析过程

1. 利用原始查询，检索出位于排序前 n 位的段落
2. 对于段落中的每一个概念 c ，计算出整个查询 q （不是单个查询词）与概念 c 之间的相似性 $\text{sim}(q, c)$ （见后面课件）
3. 根据相似性 $\text{sim}(q, c)$ 的排序，把排在前 m 位的概念添加到原始查询 q 中
 - 新加入的概念赋予一个权值 $1-0.9*i/m$
 i 表示概念 c 在最终概念排序中的位置
 m 为加入原始查询 q 中的概念数量
 - 原始查询 q 中的词项权值一般为2，以强调原始查询中词项的重要度

隐式相关反馈

- 自动全局分析 (global Analysis) 的隐式相关反馈
 - 自动局部分分析方法从检索到的结果 (局部) 文档集中提取信息以展开查询
 - 自动全局分析方法是使用来自整个文档集中的信息扩展查询

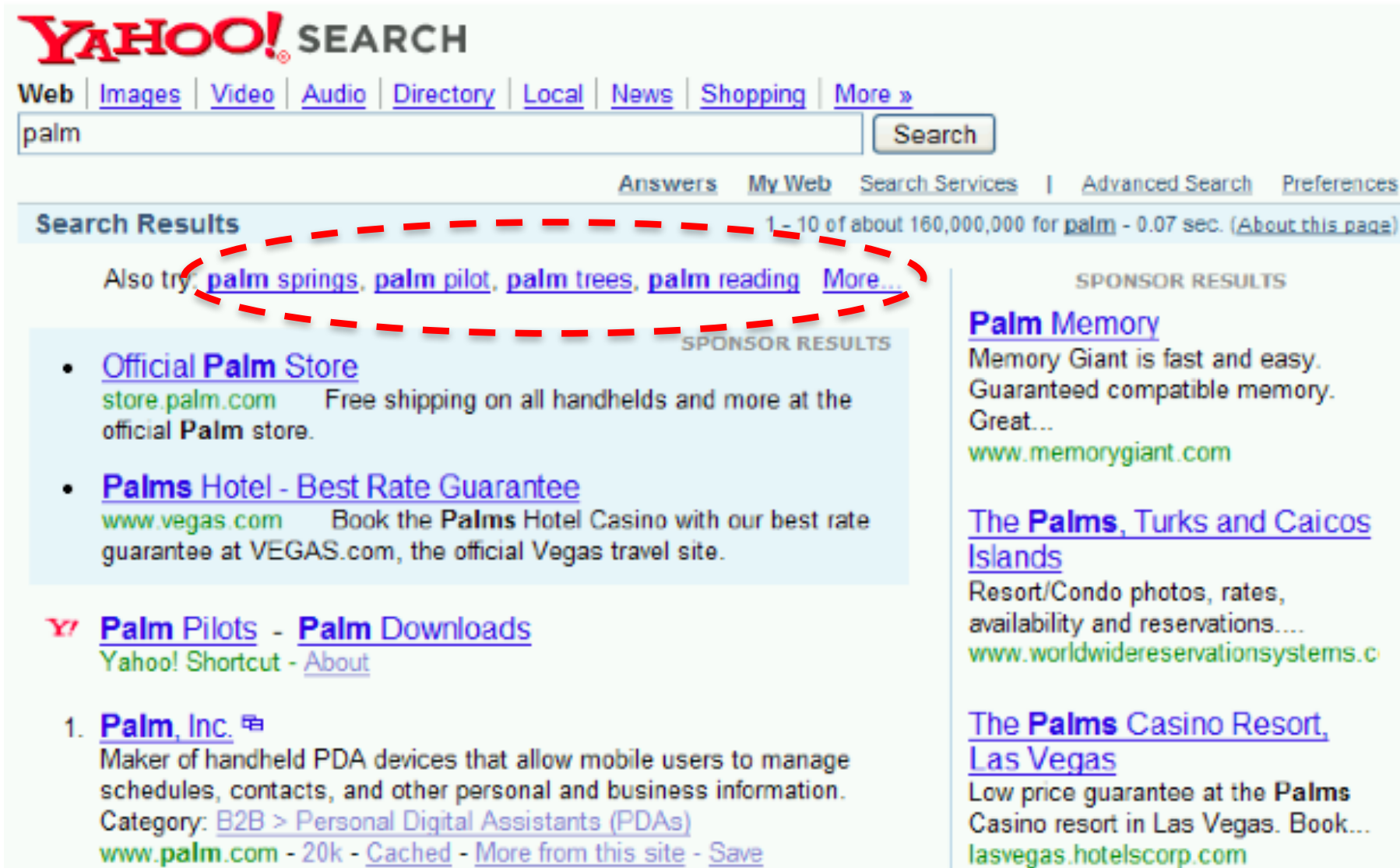
主要内容

- 相关反馈
- 查询扩展

查询扩展

- 查询扩展是另一种提高召回率的方法
- 在全局查询扩展中，查询基于一些全局的资源进行修改，这些资源是与查询无关的
 - 主要使用的信息：同义词或近义词
 - 同义词或近义词词典(thesaurus)
 - 两种同(近)义词词典构建方法：人工构建和自动构建

查询扩展



YAHOO! SEARCH

Web | Images | Video | Audio | Directory | Local | News | Shopping | More »

palm

Answers | My Web | Search Services | Advanced Search | Preferences


Search Results 1 - 10 of about 160,000,000 for palm - 0.07 sec. ([About this page](#))

Also try: [palm springs](#), [palm pilot](#), [palm trees](#), [palm reading](#) [More...](#)

SPONSOR RESULTS

- [Official Palm Store](#)
[store.palm.com](#) Free shipping on all handhelds and more at the official **Palm** store.
- [Palms Hotel - Best Rate Guarantee](#)
[www.vegas.com](#) Book the **Palms** Hotel Casino with our best rate guarantee at VEGAS.com, the official Vegas travel site.

Y! Palm Pilots - Palm Downloads
Yahoo! Shortcut - [About](#)

1. [Palm, Inc.](#) 
Maker of handheld PDA devices that allow mobile users to manage schedules, contacts, and other personal and business information.
Category: [B2B > Personal Digital Assistants \(PDAs\)](#)
[www.palm.com](#) - 20k - [Cached](#) - [More from this site](#) - [Save](#)

SPONSOR RESULTS

[Palm Memory](#)
Memory Giant is fast and easy.
Guaranteed compatible memory.
Great...
[www.memorygiant.com](#)

[The Palms, Turks and Caicos Islands](#)
Resort/Condo photos, rates, availability and reservations....
[www.worldwidereservationsystems.c](#)

[The Palms Casino Resort, Las Vegas](#)
Low price guarantee at the **Palms** Casino resort in Las Vegas. Book...
[lasvegas.hotelscorp.com](#)

查询扩展

- 人工构建的同（近）义词词典
 - 人工编辑人员维护的词典，如 PubMed
- 自动导出的同（近）义词词典
 - 例如：基于词语的共现统计信息
- 基于查询日志挖掘出的查询等价类
 - Web上很普遍，比如上面的“palm”例子

查询扩展

● 基于同（近）义词的查询扩展

- 对查询中的每个词项 t ，将词典中与 t 语义相关的词扩充到查询中
 - 例子: *hospital* → *medical*
- 通常会提高召回率，但可能会显著降低正确率
 - *interest rate* (利率) → *interest rate fascinate* (着迷)
- 广泛应用于特定领域（如科学、工程领域）的搜索引擎中
- 创建并持续维护人工词典的开销非常大

查询扩展

共现关系更加鲁棒，而语法关系更加精确

● 基于同（近）义词的查询扩展

■ 同（近）义词词典的自动构建

- 通过分析文档集中的词项分布来自动生成同（近）义词词典
- 基本的想法是计算词语之间的相似度
 - 如果两个词各自的上下文共现词类似，那么它们类似
 - “car” \approx “motorcycle”，因为它们都与“road”、“gas”及“license”之类的词共现，因此它们类似
 - 如果两个词与某些相同的词具有某种给定的语法关系的话，那么它们类似
 - harvest, peel, eat, prepare与apples 和pears有相应的语法关系，因此 apples 和pears肯定彼此类似

查询扩展

● 基于同（近）义词的查询扩展

■ 同（近）义词词典的自动构建

➤ 基于共现的词典自动构建

- 最简单的方法就是通过词典—文档矩阵 A 计算词项-词项的相似度 $C = AA^T$
- 矩阵 A 中的每个元素是词项 t 在文档 d 中加权计算后的数值：
 $w_{i,j} = (t_i, d_j)$ 的（归一化）权重
- 对每个 t_i ，选择 C 中高权重的词项进行扩展

如果矩阵 A 是0/1矩阵，那么 C 的每一项是什么？

查询扩展

● 基于同（近）义词的查询扩展

■ 同（近）义词词典的自动构建

➤ 基于共现的词典自动构建样例

| 词语 | 同(近)义词 |
|---|--|
| absolutely bottomed captivating doghouse makeup mediating keeping lithographs pathogens senses | absurd whatsoever totally exactly nothing dip copper drops topped slide trimmed shimmer stunningly superbly plucky witty dog porch crawling beside downstairs repellent lotion glossy sunscreen skin gel reconciliation negotiate case conciliation hoping bring wiping could some would drawings Picasso Dali sculptures Gauguin toxins bacteria organisms bacterial parasite grasp psyche truly clumsy naive innate |

查询扩展

- 《同义词词林》

- 梅家驹、竺一鸣、高蕴琦、殷鸿翔，上海辞书出版社1983（第一版），1996年（第二版）

- 架构

- 12大类，94中类，1428小类，3925个词群

- A人，B物，C时间和空间，D抽象事物

- E特征，F动作，G心理活动，H活动

- I现象与状态，J关联，K助语，L敬语

- 举例：

- “苹果” Bh07，“香蕉” Bh07，“西红柿”

- Bh06，……

查询扩展

- 词典的使用：举例

- 五个词的 *term-term* 相关矩阵 $A, B, C, D, \text{ and } E$.

相应的 **term** 关联

| Original term | Associated terms |
|------------------|---------------------|
| A | B |
| B | A,D |
| C | E |
| D | B,E |
| E | C,D |

$$\begin{matrix} A \\ B \\ C \\ D \\ E \end{matrix} \begin{pmatrix} A & B & C & D & E \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \end{pmatrix}$$

查询扩展

● 词典的使用：举例

- 通过增加相关词来扩展 $query$ ，相关 $term$ 的权值系数为0.5

$$q = \begin{pmatrix} A=4 \\ B=2 \\ C=1 \\ D=1 \\ E=0 \end{pmatrix} \quad \begin{array}{l} \text{Add } B=2 \\ \text{Add } A=1, D=1 \\ \text{Add } E=0.5 \\ \text{Add } B=0.5, E=0.5 \\ \text{Add nothing} \end{array}$$

$$q' = \begin{pmatrix} A=5 \\ B=4.5 \\ C=1 \\ D=2 \\ E=1 \end{pmatrix}$$

| Original term | Associated terms |
|---------------|------------------|
| A | B |
| B | A,D |
| C | E |
| D | B,E |
| E | C,D |

根据原始 $query$ 不能找出仅包含E的文档，但新的 $query$ 可以

查询扩展

● 搜索引擎中给的查询扩展

- 搜索引擎进行查询扩展主要依赖的资源：查询日志(query log)
- 例1：提交查询 [herbs] (草药)后，用户常常搜索[herbal remedies] (草本疗法)
 - → “herbal remedies” 是 “herb”的潜在扩展查询
- 例2：用户搜索 [flower pix] 时，常常点击URL photobucket.com/flower，而用户搜索 [flower clipart] 也常常点击同样的URL
 - → “flower clipart”和“flower pix”可能互为扩展查询

本章小结

- 掌握查询操作中相关反馈的作用
- 掌握查询扩展的基本方法和思想

隐式相关反馈

● 自动局部分析 (local Analysis) 的隐式相关反馈

■ 局部上下文分析

➤ 每个相关的概念 c 和原始查询 q 之间的相似度

$$sim(q, c) = \prod_{k_i \in q} \left(\delta + \frac{\log(f(c, k_i) \times idf_c)}{\log n} \right)^{idf_i}$$

$$f(c, k_i) = \sum_{j=1}^n pf_{i,j} \times pf_{c,j}$$

- N 表示排在前面的段落数
- δ 是一个比较小的因子，接近于 0.1，避免 $sim(q, c)$ 为 0
- $f(c, k_i)$ 量化了概念 c 和查询词 k_i 之间的相关性
 - $pf_{i,j}$ 表示词项 k_i 在第 j 个段落中出现的频率
 - $pf_{c,j}$ 表示词项 c 在第 j 个段落中出现的频率

隐式相关反馈

● 自动局部分析 (local Analysis) 的隐式相关反馈

■ 局部上下文分析

➤ 倒置文档频率的计算

$$idf_i = \max(1, \frac{\log_{10} N / np_i}{5})$$

$$idf_c = \max(1, \frac{\log_{10} N / np_c}{5})$$

- N 为文档集中的段落数
- np_i 为包含词项 ki 的段落数
- np_c 为包含概念 c 的段落数

➤ idf 用于强调那些不常见的查询词

隐式相关反馈

- 自动局部分析 (local Analysis) 的隐式相关反馈

- 局部上下文分析

- 以上相似度 $sim(q, c)$ 的计算采用关于 $tf-idf$ 接近的度量方法
 - 对于 $TREC$ 文档集中的数据，它对操作进行了一些调整，但对不同集合的效果不是很好
 - 因此，针对不同的文档集合，对操作进行适当的调整是非常重要的