

# 信息检索导论

## An Introduction to Information Retrieval

### 第8讲 检索评价&结果摘要

### Evaluation & Snippets

授课人：李波

中国科学院信息工程研究所/国科大网络空间安全学院

# 提纲

- ① 上一讲回顾
- ② 有关检索评价
- ③ 评价指标
- ④ 相关评测
- ⑤ 结果摘要

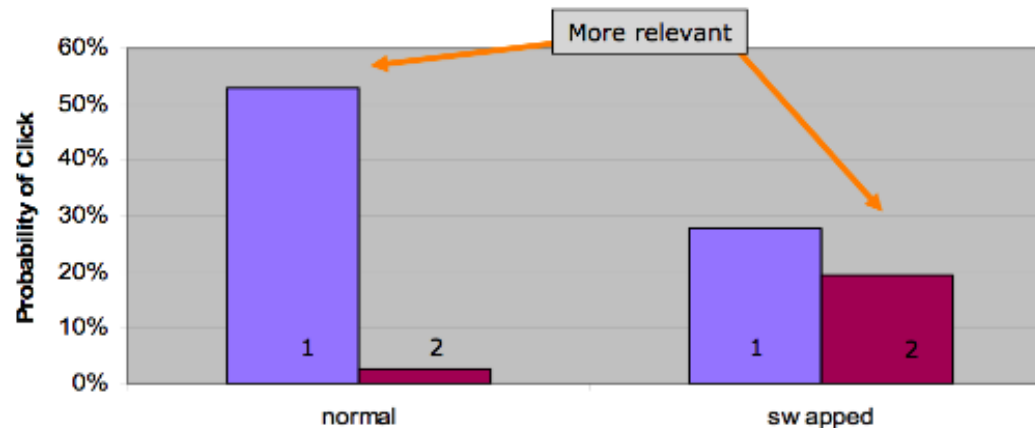
# 提纲

- ① 上一讲回顾
- ② 有关检索评价
- ③ 评价指标
- ④ 相关评测
- ⑤ 结果摘要

# 结果显示顺序对行为的影响

## Presentation bias – reversed results

- Order of presentation influences where users look **AND** where they click

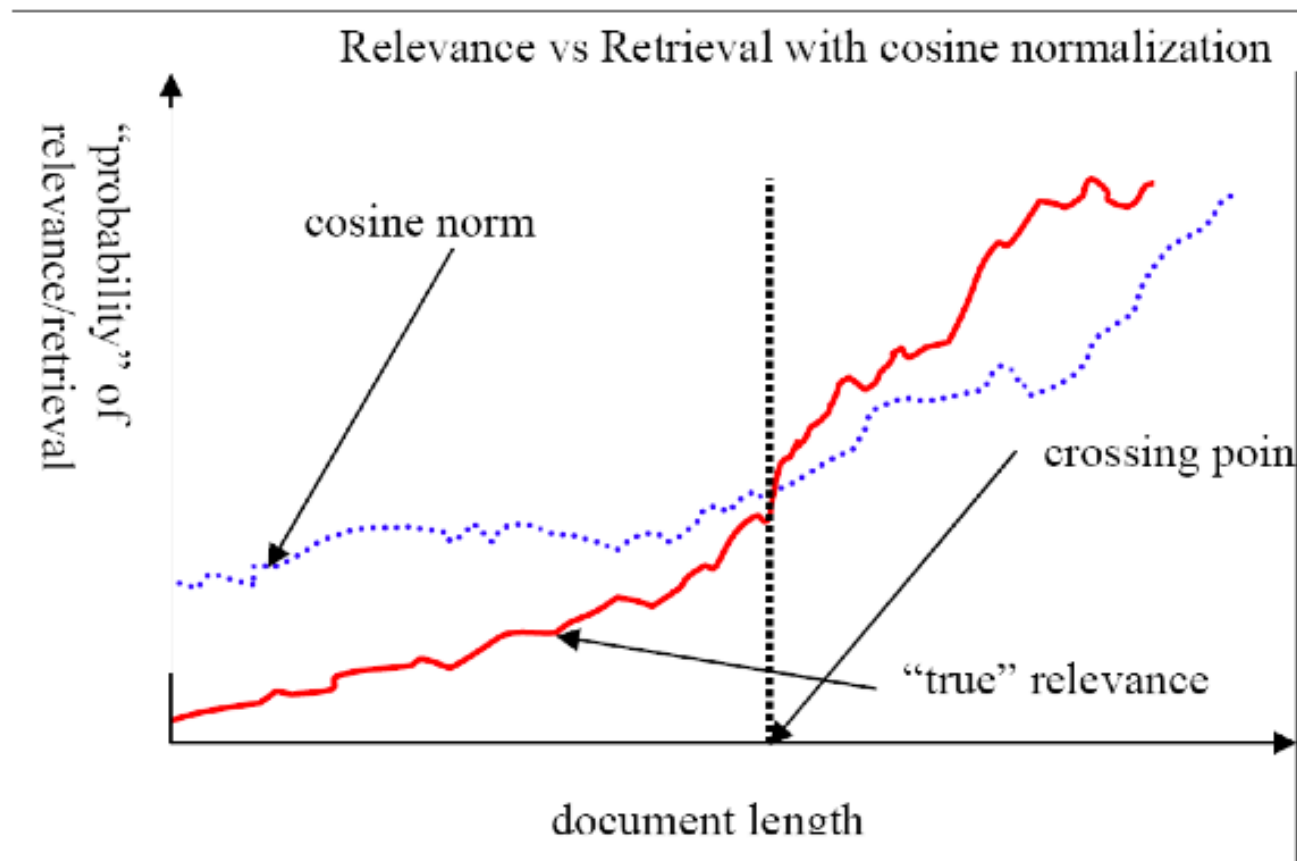


# 排序的重要性: 小结

---

- 摘要阅读(Viewing abstracts): 用户更可能阅读前几篇(1, 2, 3, 4)结果的摘要
- 点击(Clicking): 点击的分布甚至更有偏向性
  - 一半情况下, 用户点击排名最高的页面
  - 即使排名最高的页面不相关, 仍然有30%的用户会点击它。
- → 正确排序相当重要
- → 排对最高的页面非常重要

# 预测相关性概率 vs. 真实相关性概率



source: Lilian Lee

# 倒排记录表文档排序方法

- 按照docID排序
- 按与查询无关的静态得分（如PageRank）排序
  - $\text{net-score}(q, d) = g(d) + \cos(q, d)$
- 按影响度排序
  - 如对词频高的文档更感兴趣：按词频  $\text{tf}_{t,d}$  排序
- 问题：
  - 对与上述不同排序方法，哪些能保证文档在倒排记录表中的排序是一致的？

# 快速返回top $K$ 结果的启发式方法

- 以文档为单位(Document-at-a-time) 的处理
  - 计算查询-文档相似度时，先计算完文档  $d_i$  的得分，再开始文档  $d_{i+1}$  的计算
  - 文档在所有倒排记录表中的顺序应该保持一致
- 以词项为单位(Term-at-a-time)的处理
  - 计算查询-文档相似度时，先处理完词项  $t_i$  的倒排记录表，再处理词项  $t_{i+1}$  的倒排记录表
  - 需要对每个未处理完的文档建立一个累加器



# 精确top $k$ 检索及其加速办法

- 目标：从文档集的所有文档中找出 $k$ 个离查询最近的文档
- (一般)步骤：
  - 对每个文档评分(余弦相似度)
  - 按照评分高低排序
  - 选出前 $k$ 个结果
- 如何加速：
  - 思路一：加快每个余弦相似度的计算
  - 思路二：不对所有文档的评分结果排序而直接选出Top  $k$ 篇文档
  - 思路三：能否不需要计算所有 $N$ 篇文档的得分？

# 精确top $k$ 检索及其加速办法

- 方法1：快速计算余弦
  - 查询较短
  - 不考虑查询词项的权重
- 方法2：堆法N中选 $k$ 
  - 堆构建：构建包含 $k$ 个节点的堆需要 $O(J \log k)$ 次操作
  - 如果  $J=1\text{M}$ ,  $k=100$ , 那么代价大概是全部排序代价的1/3
- 方法3：提前终止计算
  - $\text{net-score}(q, d) = g(d) + \cos(q, d)$

# 堆方法实现思路

- 目标：在每一步过程中，保留top k个文档
- 使用最小堆法
- 针对当前分值为 $s'$ 的文档 $d'$ :
  - 从堆中找出当前最小分值 $h_m$  ( $O(1)$ )
  - 如果  $s' \leq h_m$ ，忽略该文档
  - 如果  $s' > h_m$ 
    - Heap-delete-root ( $O(\log k)$ )
    - Heap-add  $d'/s'$  ( $O(\log k)$ )
- 输出k个文档( $O(k \log k)$ )
  - k次Heap-delete-root

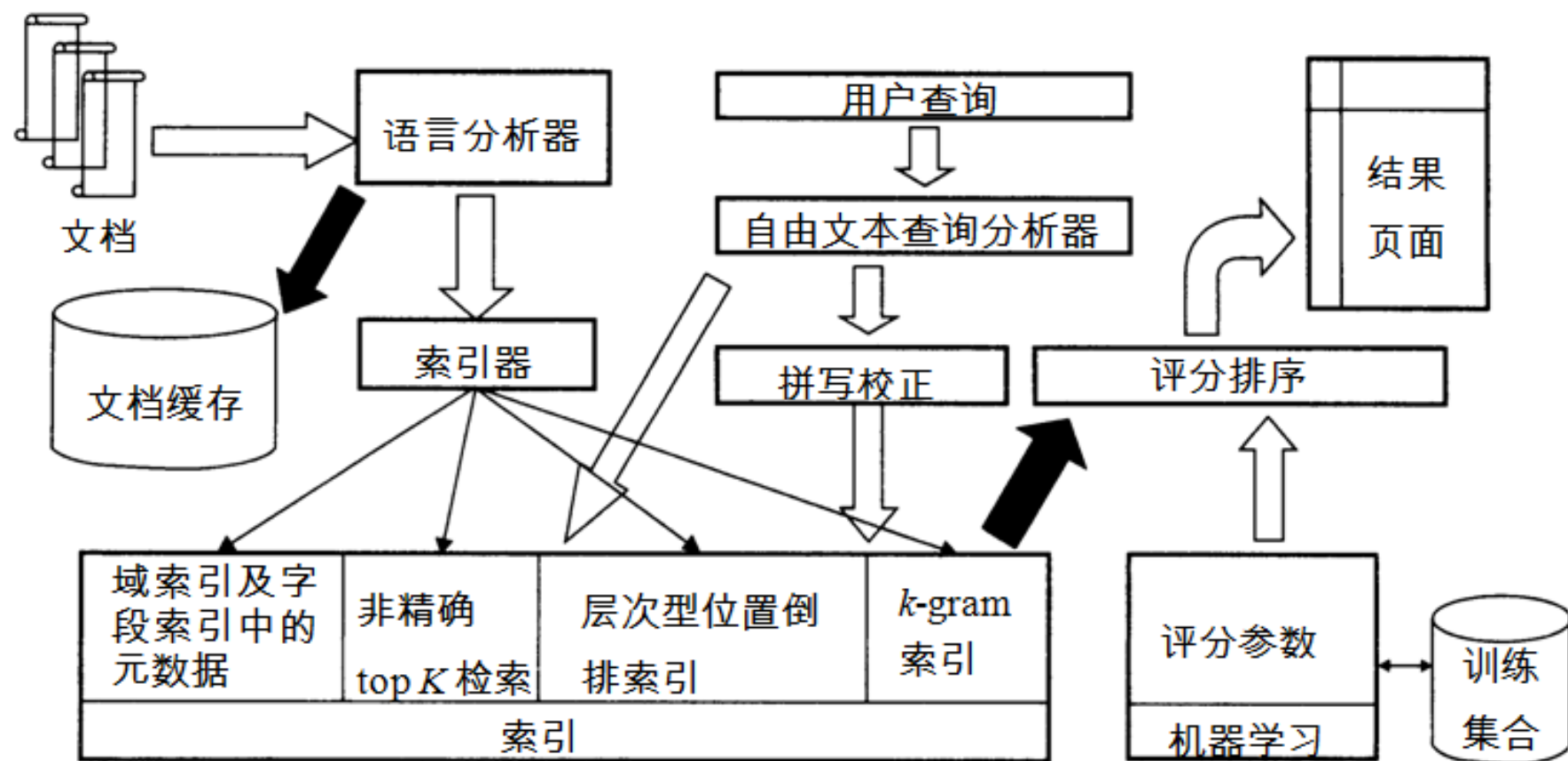
# 非精确top $k$ 检索

- 找一个文档集合A,  $k < |A| \ll N$ , 利用A中的top  $k$  结果代替整个文档集的top  $k$  结果
  - 即给定查询后, A是整个文档集上近似剪枝得到的结果
- 方法1: 索引去除(Index elimination)
  - 只考虑那些包含高idf查询词项的文档
  - 只考虑那些包含多个查询词项的文档(比如达到一定比例, 3个词项至少出现2个, 4个中至少出现3个等等)
- 方法二: 胜者表(Champion list)
  - 对每个词项 $t$ , 预先计算出其倒排记录表中权重最高的 $r$ 篇文档, 如果采用tfidf机制, 即tf最高的 $r$ 篇
  - 这 $r$ 篇文档称为 $t$ 的胜者表

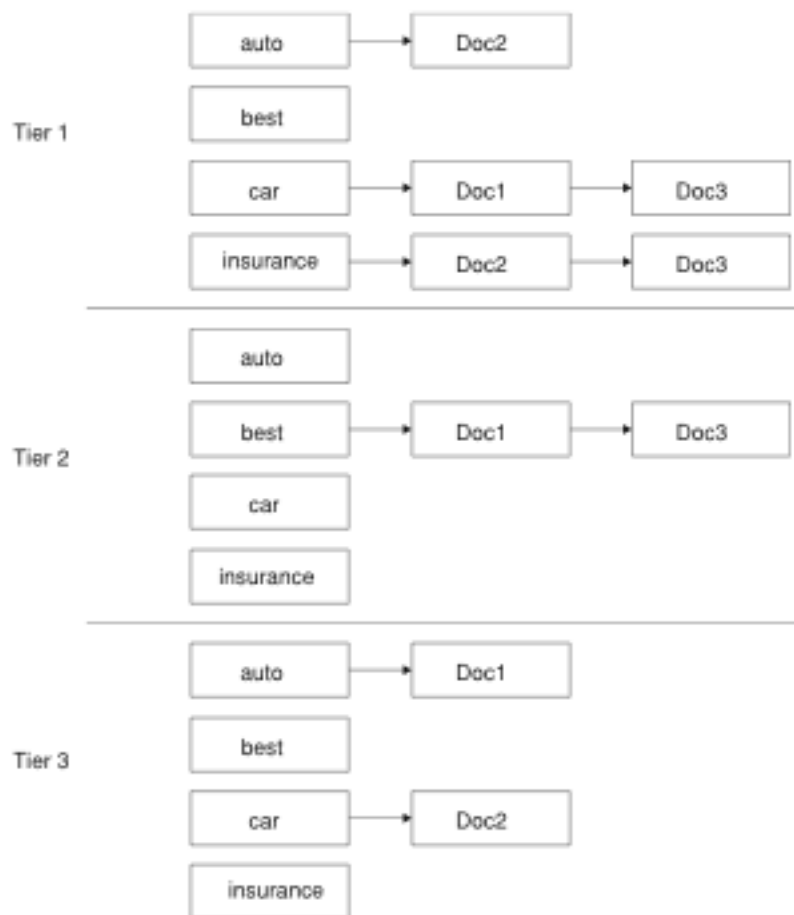
# 非精确top $k$ 检索（续）

- 方法3：静态质量得分排序方式（net-score最高的top  $k$ 文档）
  - 全局胜者表：选择 $r$ 值，对每个词项维护一张胜者表，该表中放置了 $r$ 篇 $g(d) + \text{tf-idf}_{t,d}$ 值最高的文档
  - 高端表(High list)和低端表(Low list)：按 $\text{tf}$ 划分两个无交集的倒排记录表——层次型索引
- 方法4：影响度(Impact)排序——以词项为单位计算
  - 影响度：按 $\text{tf}$ 排序，按 $g(d)$ 排序等
  - 对于多词项组成的查询，按照idf从大到小扫描词项
  - 提前结束法+忽略低idf词项法
- 方法5：簇剪枝(Cluster pruning)
  - $\sqrt{N}$ 个先导者+  $\sqrt{N}$ 个追随者

# 完整的搜索系统示意图



# 分层索引



- 第1层：所有 $tf > 20$ 的索引
- 第2层：所有 $tf > 10$ 的索引
- 第3层：所有 $tf < 10$ 的索引

# 域索引

- 对域构建倒排索引
  - $(term, docID) \rightarrow (term, field, docID)$
- 一种倒排索引的形式为:

William.author	2	4	8	16	32	64
William.title	1	2	3	5	8	13
William.abstract	1	3	5	7	9	11

- 课程所介绍的压缩和排序方法适用于域索引



# 本讲内容

---

- 信息检索的评价指标
  - 不考虑序的检索评价指标(即基于集合)
  - 考虑序的评价指标
- 信息检索评测语料及会议
- 检索结果的摘要

# 提纲

- ① 上一讲回顾
- ② 有关检索评价
- ③ 评价指标
- ④ 相关评测
- ⑤ 结果摘要

# 关于评价

---

- 评价无处不在，既重要也很必要
  - 工作、生活、娱乐、找对象、招生
- 评价很难，但是似乎又很容易
  - 人的因素、标准、场景
- 评价是检验学术进步的唯一标准，也是杜绝学术腐败的有力武器

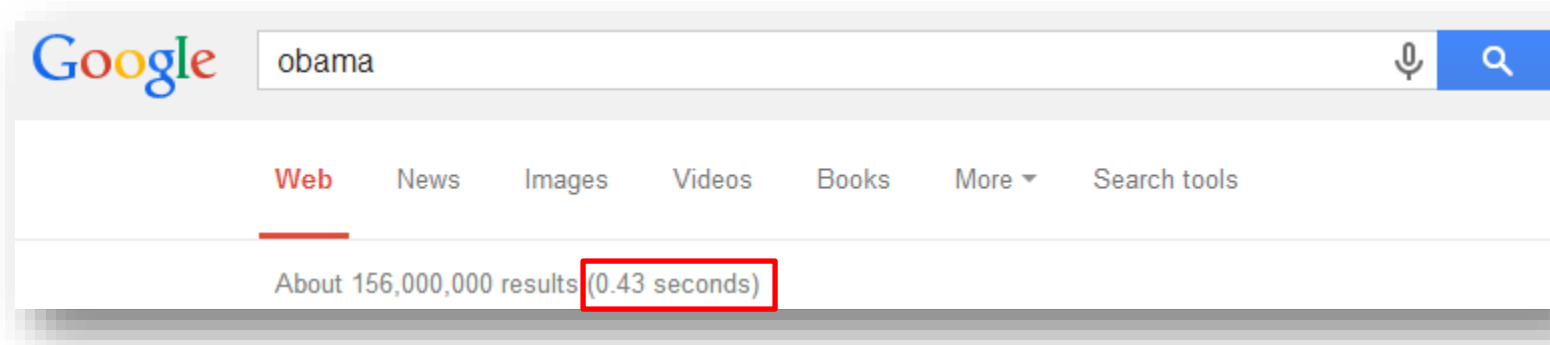
# 为什么要评价IR？

---

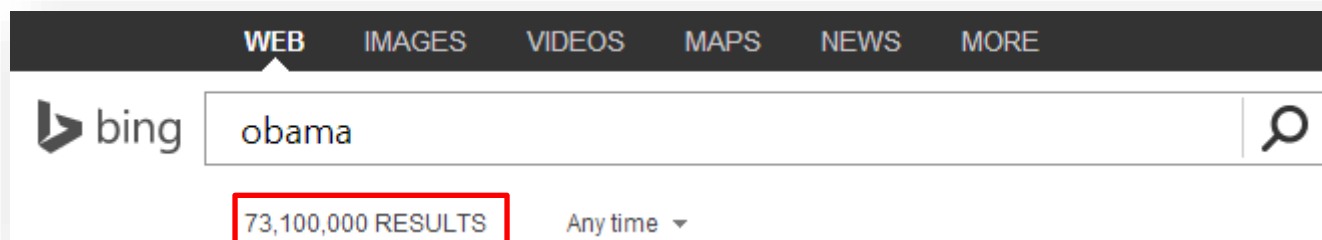
- 通过评估可以评价不同技术的优劣，不同因素对系统的影响，从而促进本领域研究水平的不断提高
  - 类比：110米栏各项技术---起跑、途中跑、跨栏、步频、冲刺等等
- 信息检索系统的目标是较少消耗情况下尽快、全面返回准确的结果。

# 哪个搜索引擎更好： Bing vs. Google?

- 评价标准是什么？
  - 查询请求的响应时间是否足够快？

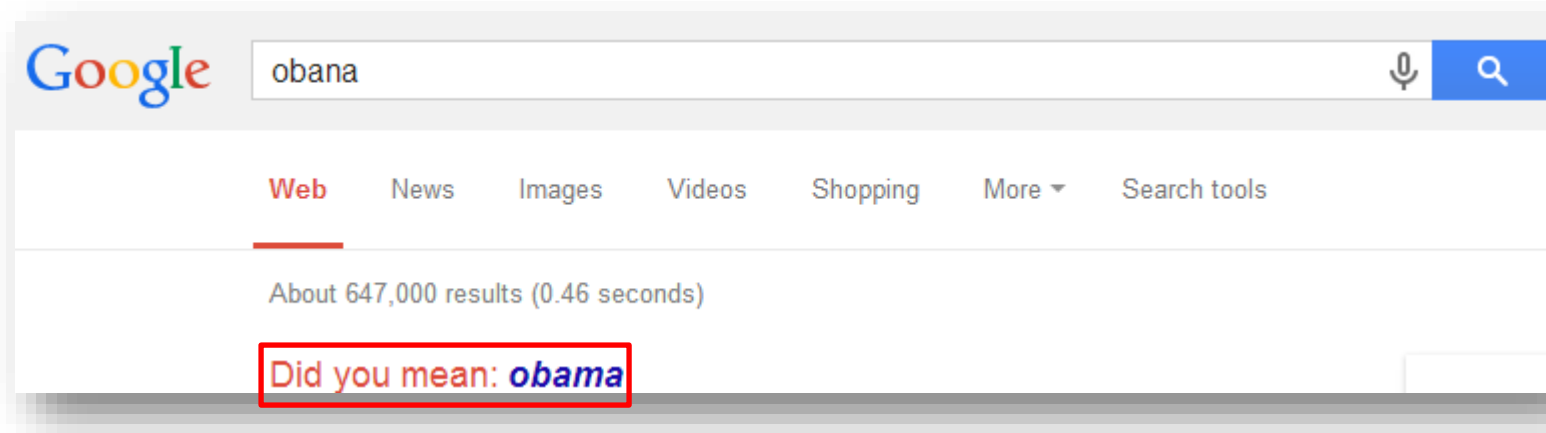


- 返回文档数的多少？（或索引的文档数更多？）



# 哪个搜索引擎更好：Bing vs. Google?

- 评价标准是什么？
  - 是否能对输入的拼写错误进行校正？



- 是否能推荐好的相关查询？



# IR中评价什么？

---

- 效率 (Efficiency)—可以采用通常的评价方法
  - 时间开销
  - 空间开销
  - 响应速度
- 效果 (Effectiveness)
  - 返回的文档中有多少相关文档
  - 所有相关文档中返回了多少
  - 返回得靠不靠前
- 其他指标
  - 覆盖率(Coverage)
  - 访问量
  - 数据更新速度

# IR中评价什么？

---

- “*how well a system meets the information needs of its users.*” – *wiki*
- 用户的满意度的体现
  - 结果页面的点击量增加
  - 结果页面的访问量/下单量增加
  - 返回结果与需求的相关性



# 如何评价效果/相关性？

- 相关性评价需要的3个要素
  - 一个文档集
  - 一组用于测试信息需求的集合，通常表示为查询请求
  - 一组相关性判定结果，说明每个文档和查询请求的关系
    - 相关 or 不相关
  
- 相同的文档集合，相同的查询主题集合，相同的评价指标，不同的检索系统进行比较
  - **The Cranfield Experiments**, Cyril W. Cleverdon, 1957 –1968 (上百篇文档集合)
  - **SMART System**, Gerald Salton, 1964-1988 (数千篇文档集合)
  - **TREC(Text REtrieval Conference)**, Donna Harman, 美国标准技术研究所, 1992 - (上百万篇文档)，信息检索的“奥运会”

# 评价任务的例子

- 两个系统，一批查询，对每个查询每个系统分别得到一些结果。目标：哪个系统好？

系统&查询	1	2	3	4	...
系统1， 查询1	d3	d6	d8	d10	
系统1， 查询2	d1	d4	d7	d11	
系统2， 查询1	d6	d7	d3	d9	
系统2， 查询2	d1	d2	d4	d13	

# 评价的关键环节

---

- 评价指标：某个或某几个可衡量、可比较的值
- 评价过程：设计上保证公平、合理

# 提纲

- ① 上一讲回顾
- ② 有关检索评价
- ③ 评价指标
- ④ 相关评测
- ⑤ 结果摘要

# 评价指标分类

---

- 对单个查询进行评估的指标
  - 在单个查询上检索系统的得分
- 对多个查询进行评估的指标
  - 在多个查询上检索系统的得分

# 评价指标分类

---

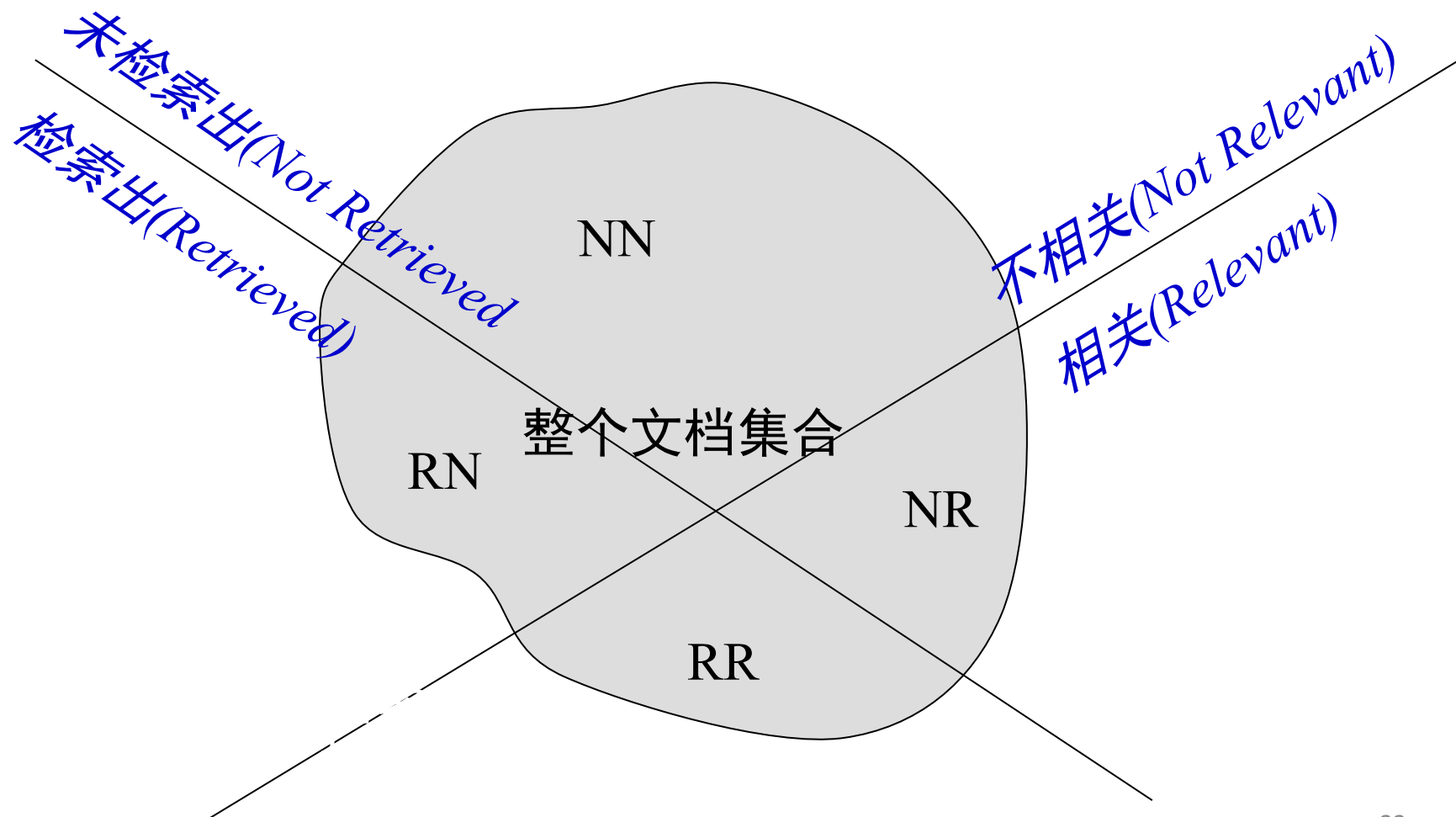
- 对单个查询进行评估的指标 ←
  - 在单个查询上检索系统的得分
- 对多个查询进行评估的指标
  - 在多个查询上检索系统的得分

# 回到例子

系统&查询	1	2	3	4	...
系统1, 查询1	d3 ✓	d6 ✓	d8	d10	
系统1, 查询2	d1	d4	d7	d11	
系统2, 查询1	d6 ✓	d7	d2	d9 ✓	
系统2, 查询2	d1	d2	d4	d13	

查询1的标准答案集合 {d3,d4,d6,d9}

# 整个文档集合的划分





# 评价指标

- 召回率(Recall):  $RR/(RR + NR)$ , 返回的相关结果数占实际相关结果总数的比率, 也称为**查全率**,  $R \in [0,1]$
- 正确率(Precision):  $RR/(RR + RN)$ , 返回的结果中真正相关结果的比率, 也称为**查准率**,  $P \in [0,1]$
- 两个指标分别度量检索效果的某个方面, 忽略任何一个方面都有失偏颇。两个极端情况: 返回有把握的1篇,  $P=100\%$ , 但 $R$ 极低; 全部文档都返回,  $R=1$ , 但 $P$ 极低

# 四种关系的矩阵表示

真正相关文档 RR+NR    真正不相关文档

系统判定相关  
RR+RN (检索出)

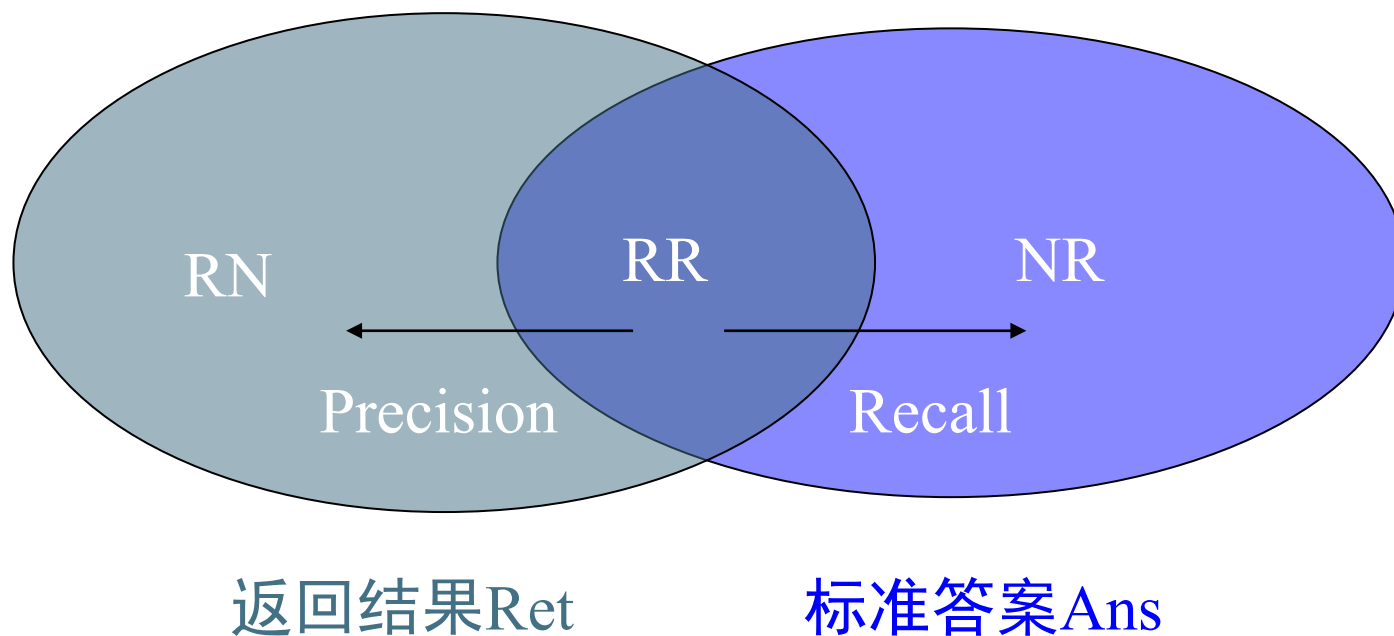
系统判定不相关  
(未检索出)

RR	RN
NR	NN

Recall

Precision

# 基于集合的图表示



# 回到例子

系统&查询	1	2	3	4	5
系统1, 查询1	d3 ✓	d6 ✓	d8	d10	d11
系统1, 查询2	d1	d4	d7	d11	d13
系统2, 查询1	d6 ✓	d7	d2	d9 ✓	/
系统2, 查询2	d1	d2	d4	d13	d14

对于查询1的标准答案集合 {d3,d4,d6,d9}

对于系统1, 查询1, 正确率2/5, 召回率2/4

对于系统2, 查询1, 正确率2/4, 召回率2/4

# 课堂提问：另一个计算例子

- 查询Q，本应该有100篇相关文档，某个系统返回200篇文档，其中80篇是真正相关的文档
  - $\text{Recall} = 80/100 = 0.8$
  - $\text{Precision} = 80/200 = 0.4$
  - 结论：召回率较高，但是正确率较低

# 正确率和召回率的应用领域

---

- 拼写校对
- 中文分词
- 文本分类
- 人脸识别
- .....

# 关于正确率和召回率的讨论(1)

- “宁可错杀一千，不可放过一人” ➔ 偏重召回率，忽视正确率。冤杀太多。
- 判断是否有罪：
  - 如果没有证据证明你无罪，那么判定你有罪。 ➔ 召回率高，有些人受冤枉
  - 如果没有证据证明你有罪，那么判定你无罪。 ➔ 召回率低，有些人逍遥法外

# 关于正确率和召回率的讨论(2)

- 虽然Precision和Recall都很重要，但是不同的应用、不同的用户可能会对两者的要求不一样。因此，实际应用中应该考虑这点。
  - 垃圾邮件过滤：宁愿漏掉一些垃圾邮件，但是尽量少将正常邮件判定成垃圾邮件。
  - 有些用户希望返回的结果全一点，他有时间挑选；有些用户希望返回结果准一点，他不需要结果很全就能完成任务。



# 课堂提问：

- 正确率和召回率的定义或者计算有什么问题或不足？

系统&查询	1	2	3	4	5
系统1， 查询1	d3 ✓	d6 ✓	d8	d10	d11
系统1， 查询2	d1	d4	d7	d11	d13
系统2， 查询1	d6 ✓	d7	d2	d9 ✓	/
系统2， 查询2	d1	d2	d4	d13	d14

对于查询1的标准答案集合 {d3,d4,d6,d9}

对于系统1， 查询1， 正确率2/5， 召回率2/4

对于系统2， 查询1， 正确率2/4， 召回率2/4

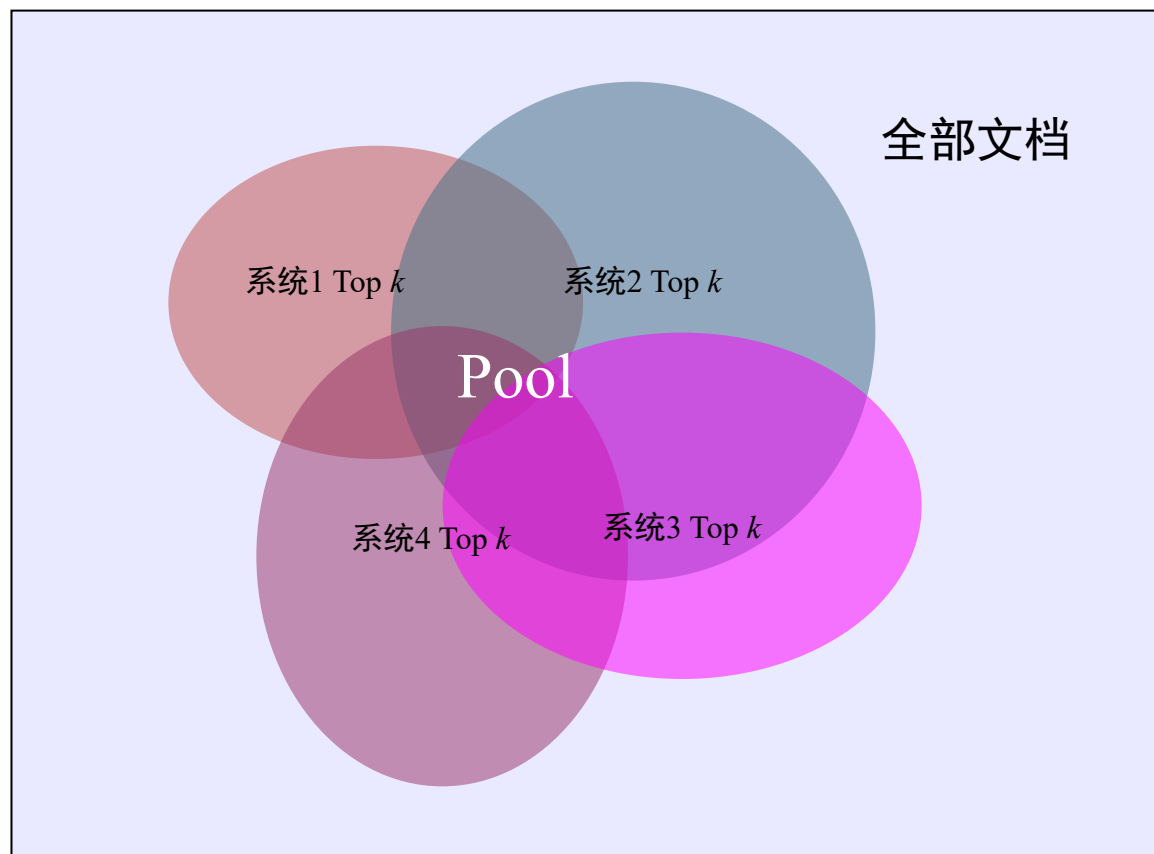
# 正确率和召回率的问题

- 召回率难以计算
  - 解决方法：Pooling方法，或者不考虑召回率
- 两个指标分别衡量了系统的某个方面，但是也为比较带来了难度，究竟哪个系统好？大学最终排名也只有一个指标。
  - 解决方法：单一指标，将两个指标融成一个指标
- 两个指标都是基于(无序)集合进行计算，并没有考虑序的作用
  - 举例：两个系统，对某个查询，返回的相关文档数目一样都是10，但是第一个系统是前10条结果，后一个系统是最后10条结果。显然，第一个系统优。但是根据上面基于集合的计算，显然两者指标一样。
  - 解决方法：引入序的作用

# 问题一：召回率难以计算

- 对于大规模语料集合，列举每个查询的所有相关文档是不可能的事情，因此，这种情况几乎不可能准确地计算召回率
- 缓冲池(Pooling)方法：对多个检索系统的Top  $k$  个结果组成的集合(并集)进行人工标注，标注出的相关文档集合作为整个相关文档集合。这种做法被验证是可行的(可以比较不同系统的相对效果)，在TREC会议中被广泛采用。

# 4个系统的Pooling



# 课堂提问

- (某个系统的某个查询)通过Pooling计算出的召回率、正确率和真正的召回率、正确率的大小之间有什么关系？
- 情况1(常见情况): 如果只有部分结果进行了Pooling操作
  - 计算正确率时有  $RR_p \leq RR$ , 分母都是Ret不变, 此时计算出来的正确率会小于真实的正确率
  - 而对于召回率, 计算中的分子分母都变小, 所以结果不一定
- 情况2: 如果所有的结果都进行了Pooling
  - 那么正确率计算时的分子分母都不变, 此时计算出的正确率等于真实的正确率
  - 由于分子不变, 而分母显然小于真实的相关文档总数, 所以计算出来的召回率大于真实的召回率。

## 问题二：P和R需要融合

- F值(F-measure): 召回率R和正确率P的调和平均值, if  $P=0$  or  $R=0$ , then  $F=0$ , else 采用下式计算:

$$F = \frac{2}{\frac{1}{P} + \frac{1}{R}} = \frac{2PR}{P+R} \quad (P \neq 0, R \neq 0)$$

- $F_\beta$ : 表示召回率的重要程度是正确率的 $\beta( \geq 0)$ 倍,  $\beta > 1$ 更重视召回率,  $\beta < 1$ 更重视正确率

$$F_\beta = \frac{(1 + \beta^2)PR}{\beta^2 P + R} \quad (P \neq 0, R \neq 0)$$

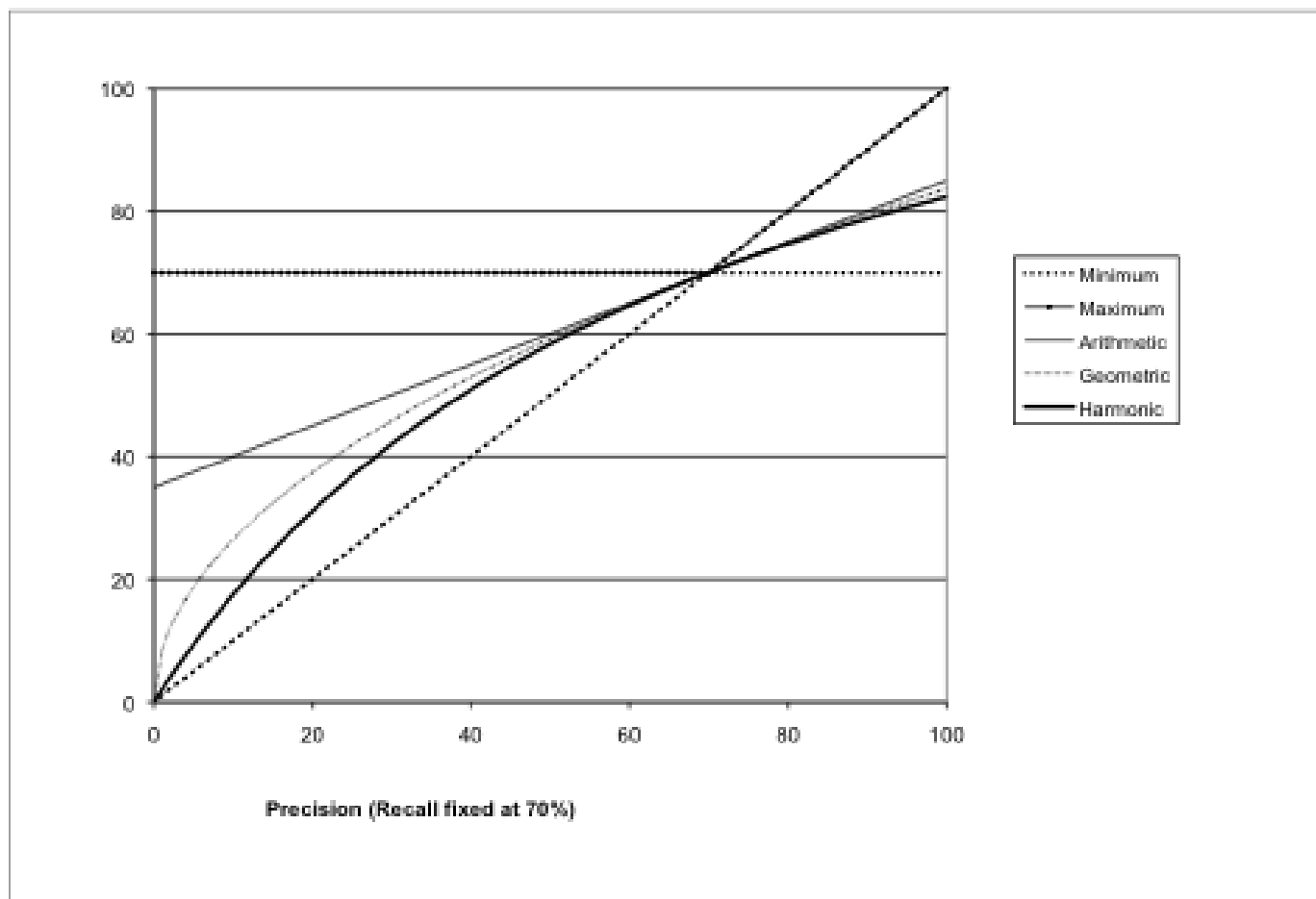
- E(Effectiveness)值: 召回率R和正确率P的加权平均值,  $b > 1$ 表示更重视P,  $E = 1 - F_\beta$ ,  $b^2 = 1/\beta^2$

$$E = 1 - \frac{1 + b^2}{\frac{b^2}{P} + \frac{1}{R}} \quad (P \neq 0, R \neq 0)$$

# 为什么使用调和平均计算F值

- 为什么不使用其他平均来计算F，比如算术平均
  - 调和平均、几何平均、算术平均
- 如果采用算术平均计算F值，那么一个返回全部文档的搜索引擎的F值就不低于50%，明显过高
  - 做法：不管是P还是R，如果十分低，那么结果应该表现出来，即这样的情形下最终的F值应该有所惩罚
  - 采用P和R中的最小值可能达到上述目的
  - 但是最小值方法不平滑而且不易加权
- 基于调和平均计算出的F值可以看成是平滑的最小值函数

# $F_1$ 及其他平均计算方法





# 精确率(Accuracy)

- 精确率是所有判定中正确的比率
  - $\text{accuracy} = (\text{RR} + \text{NN}) / (\text{RN} + \text{RR} + \text{NR} + \text{NN})$
- 为什么通常使用P、R、F而不使用精确率？
- Web信息检索当中精确率为什么不可用？

# 课堂练习

- 计算P、R、F1

	相关	不相关
返回	18	2
未返回	82	1,000,000,000

- 下面的一个搜索引擎无论对于什么查询都返回0结果，为什么该引擎例子表明使用精确率是不合适的？



# 精确率不适合IR的原因

- 由于和查询相关毕竟占文档集的极少数，所以即使什么都不返回也会得到很高的精确率
- 什么都不返回可能对大部分查询来说可以得到 99.99%以上的精确率（非均衡问题！）
- 信息检索用户希望找到某些文档并且能够容忍结果中有一定的不相关性
- 返回一些即使不好的文档也比不返回任何文档强
- 因此，实际中常常使用P、R和F1，而不使用精确率

## 问题三：P、R没有考虑结果的序

- **R-Precision:** 检索结果中，在所有相关文档总数位置上的准确率，如某个查询的相关文档总数为80，则计算检索结果中在前80篇文档的正确率。

系统1， 查询1	d3✓	d6 ✓	d8	d10	d11
系统2， 查询1	d6 ✓	d7	d2	d9 ✓	

对于查询1的标准答案集合 {d3,d4,d6,d9}

系统1 R-Precision=2/4    系统2 R-Precision=2/4

# 引入序的作用

- 正确率-召回率 曲线(precision versus recall curve)
  - 检索结果以排序方式排列，用户不可能马上看到全部文档，因此，在用户观察的过程中，正确率和召回率在不断变化(vary)
  - 可以求出在召回率分别为0%,10%,20%,30%,...,90%,100%上对应的正确率，然后描出图像
  - 在上面的曲线对应的系统结果更好

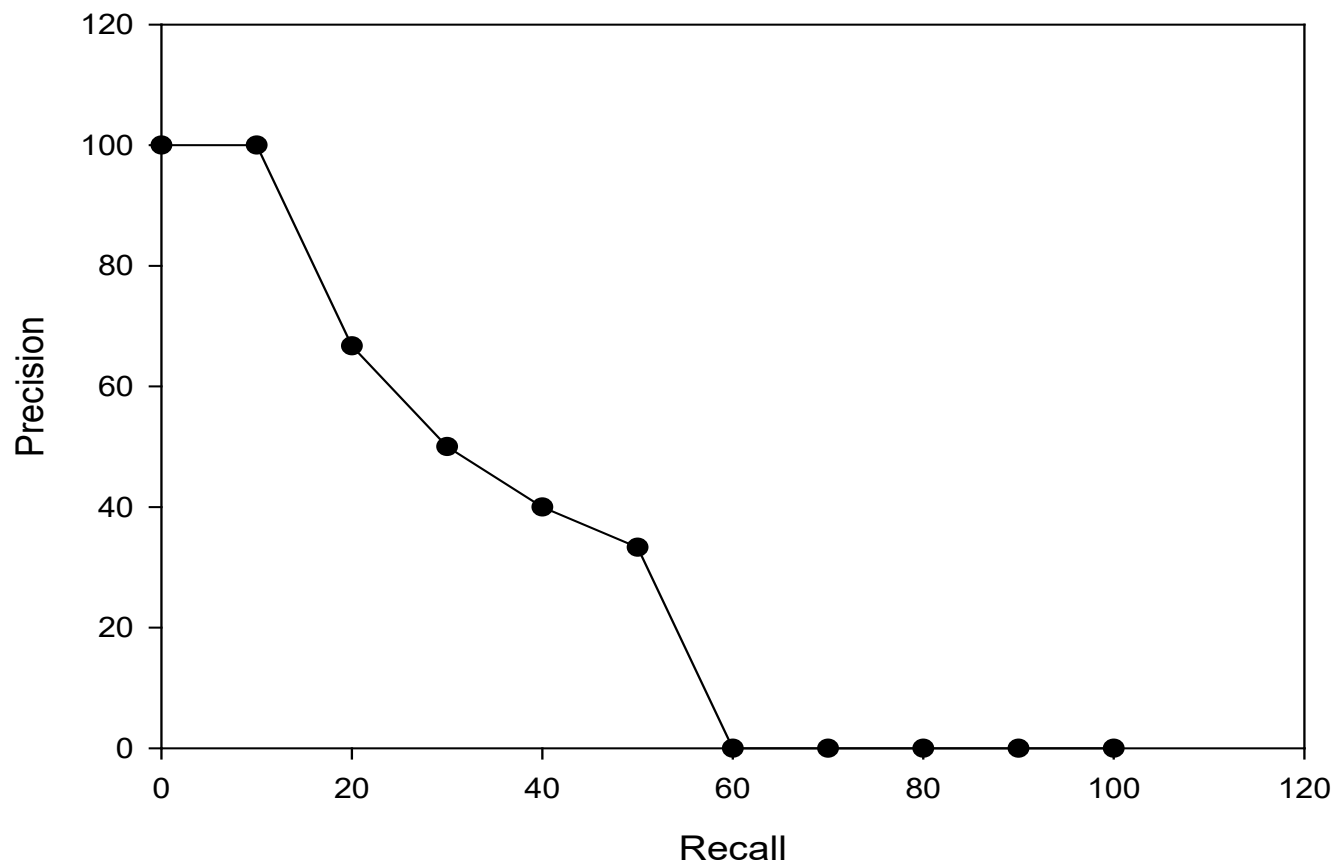
# P-R曲线的例子

- 某个查询q的标准答案集合为：  
 $R_q = \{d3, d5, d9, d25, d39, d44, d56, d71, d89, d123\}$
- 某个IR系统对q的检索结果如下：

1. d123 R=0.1,P=1	6. d9 R=0.3,P=0.5	11. d38
2. d84	7. d511	12. d48
3. d56 R=0.2,P=0.67	8. d129	13. d250
4. d6	9. d187	14. d113
5. d8	10. d25 R=0.4,P=0.4	15. d3 R=0.5,P=0.33

# P-R曲线

Precision-recall 曲线

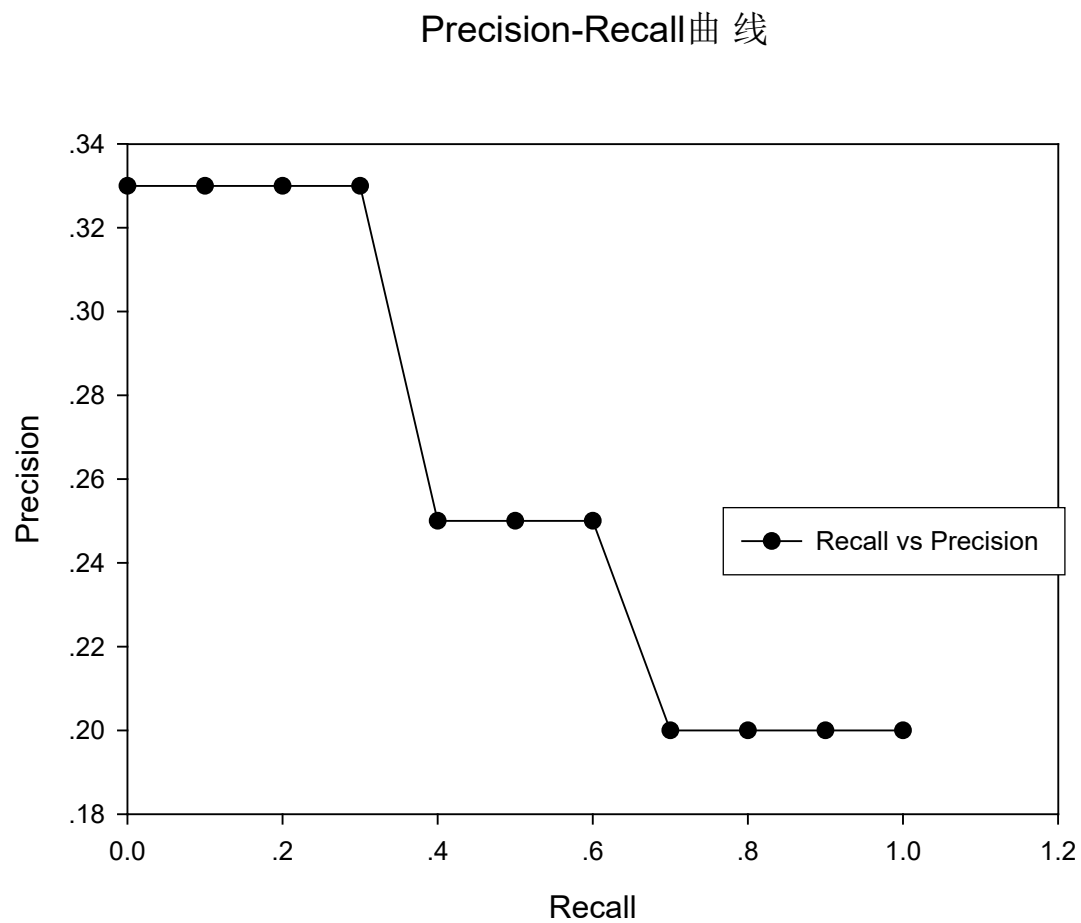


# P-R 曲线的插值问题

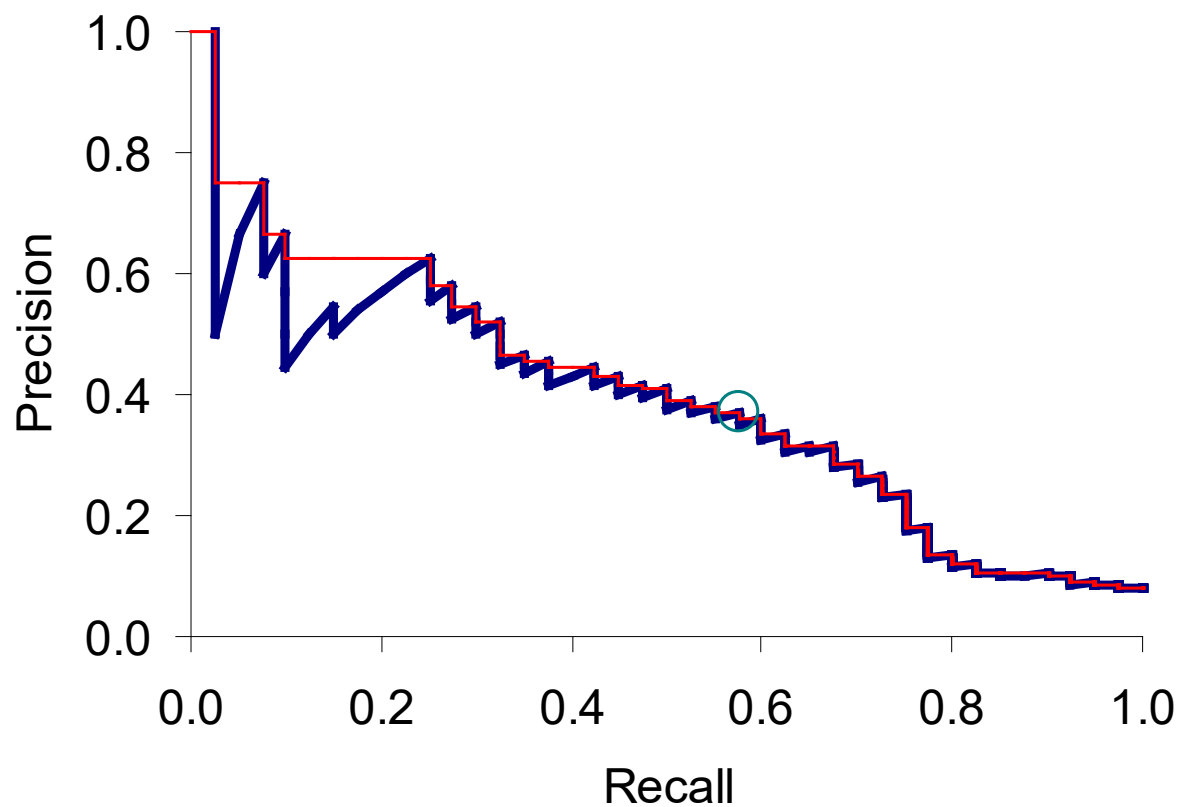
- 对于前面的例子，假设 $R_q = \{d3, d56, d129\}$ 
  - 3. d56  $R=0.33, P=0.33$
  - 8. d129  $R=0.66, P=0.25$
  - 15. d3  $R=1, P=0.2$
- 不存在10%, 20%, ..., 90%的召回率点，而只存在33.3%, 66.7%, 100%三个召回率点
- 在这种情况下，需要利用存在的召回率点对不存在的召回率点进行插值(interpolate)
- 对于召回率 $t\%$ ，如果不存在该召回率点，则定义 $t\%$ 为从 $t\%$ 到 $(t+10)\%$ 中最大的正确率值。
- 对于上例，0%, 10%, 20%, 30%上正确率为0.33，40%~60%对应0.25，70%以上对应0.2



# P-R曲线图



# P-R曲线图



# P-R曲线的优缺点

---

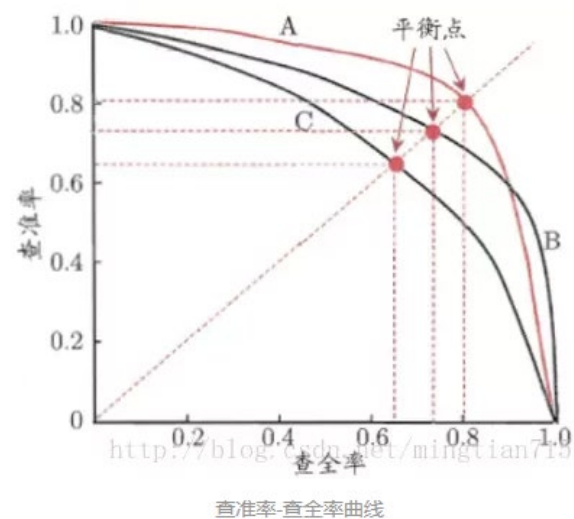
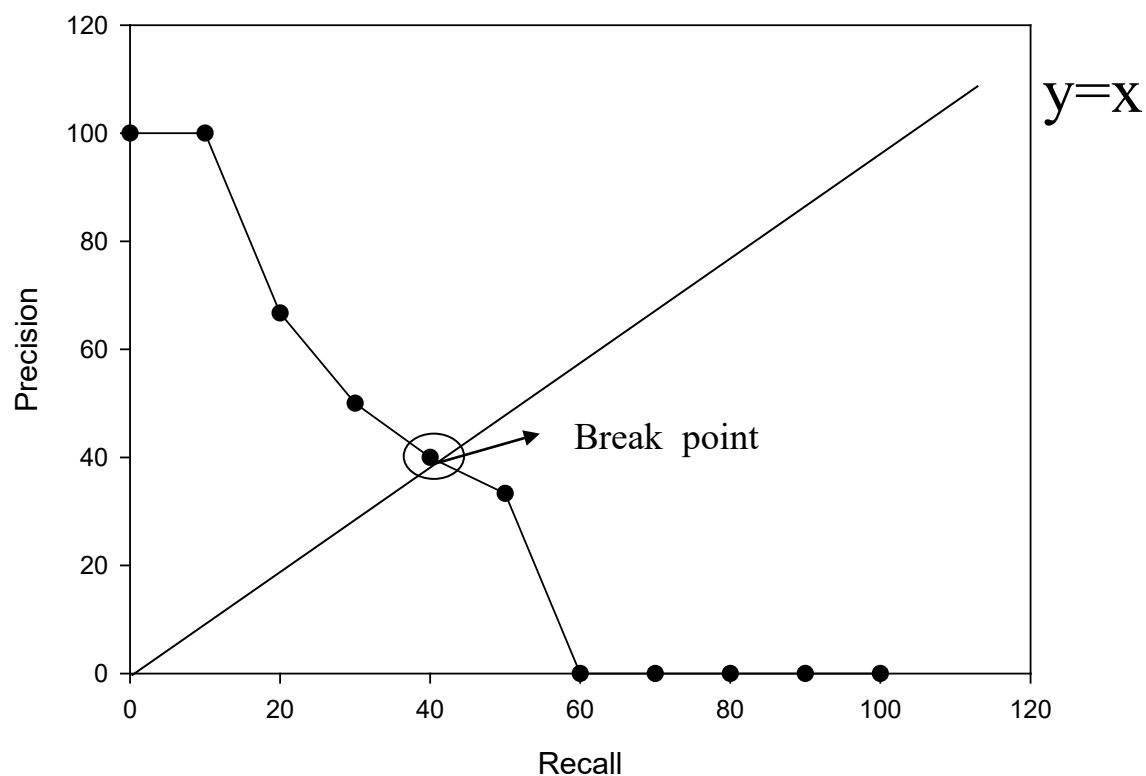
- 优点：
  - 简单直观
  - 既考虑了检索结果的覆盖度，又考虑了检索结果的排序情况
- 缺点：
  - 单个查询的P-R曲线虽然直观，但是难以明确表示两个系统的优劣

# 基于P-R曲线的单一指标

- Break Point: P-R曲线上  $P=R$  的那个点
  - 这样可以直接进行单值比较多个系统的优劣
- 11点平均正确率(11 point average precision): 在召回率分别为0,0.1,0.2,...,1.0的十一个点上的正确率求平均, 等价于插值的AP

# P-R曲线中的break point

Precision-recall 曲线



# 平均正确率

- 平均正确率(Average Precision, AP): 对不同召回率点上的正确率进行平均
  - 未插值的AP(常用): 某个查询Q共有6个相关结果, 某系统排序返回了5篇相关文档, 其位置分别是第1, 第2, 第5, 第10, 第20位, 则
$$AP=(1/1+2/2+3/5+4/10+5/20+0)/6$$
  - 插值的AP: 在召回率分别为0,0.1,0.2,...,1.0的十一个点上的正确率求平均, 等价于11点平均
  - 只对返回的相关文档进行计算的AP,
$$AP=(1/1+2/2+3/5+4/10+5/20)/5$$
, 倾向那些快速返回结果的系统, 没有考虑召回率

# 不考虑召回率

- Precision@N: 在前N个结果上的正确率
  - 对于搜索引擎，大量统计数据表明，大部分搜索引擎用户只关注前一、两页的结果，因此， $P@10$ ,  $P@20$  对大规模搜索引擎来说是很好的评价指标
- NDCG: 后面详细介绍

# 回到例子

系统&查询	1	2	3	4	5
系统1, 查询1	d3 ✓	d6 ✓	d8	d10	d11
系统1, 查询2	d1 ✓	d4	d7	d11	d13 ✓
系统2, 查询1	d6 ✓	d7	d2	d9 ✓	/
系统2, 查询2	d1 ✓	d2 ✓	d4	d13 ✓	d14

查询1及查询2的标准答案集合分别为  $\{d3, d4, d6, d9\}$   $\{d1, d2, d13\}$ ，求 $P@2, P@5$

系统1查询1:  $P@2=1$ ,  $P@5=2/5$ ; 系统1查询2:  $P@2=1/2$ ,  $P@5=2/5$ ;

系统2查询1:  $P@2=1/2$ ,  $P@5=2/5$ ; 系统2查询2:  $P@2=1$ ,  $P@5=3/5$

系统1查询1:  $AP=?$  系统1查询2:  $AP=?$

系统2查询1:  $AP=?$  系统2查询2:  $AP=?$



# 评价指标分类

---

- 对单个查询进行评估的指标
  - 对单个查询得到一个结果
- 对多个查询进行评估的指标 ←
  - 在多个查询上检索系统的得分求平均

# 评价指标

- 平均的求法：
  - 宏平均(Macro Average): 对每个查询求出某个指标，然后对这些指标进行算术平均
  - 微平均(Micro Average): 将所有查询视为一个查询，将各种情况的文档总数求和，然后进行指标的计算
    - 如：Micro Precision=(对所有查询检出的相关文档总数)/(对所有查询检出的文档总数)
  - 宏平均对所有查询一视同仁，微平均受返回相关文档数目比较大的查询影响(宏平均保护弱者，类比：乒乓球参赛资格限制)
- MAP(Mean AP): 对所有查询的AP求宏平均

# 回到例子

系统&查询	1	2	3	4	5
系统1, 查询1	d3 ✓	d6 ✓	d8	d10	d11
系统1, 查询2	d1 ✓	d4	d7	d11	d13 ✓
系统2, 查询1	d6 ✓	d7	d2	d9 ✓	/
系统2, 查询2	d1 ✓	d2 ✓	d4	d13 ✓	d14

查询1及查询2的标准答案集合分别为  $\{d3, d4, d6, d9\}$   $\{d1, d2, d13\}$ , 求宏平均/微平均/MAP

系统1查询1:  $P=2/5$ ,  $R=2/4$ ,  $F=4/9$ ,  $AP=1/2$ ; 系统1查询2:  $P=2/5$ ,  $R=2/3$ ,  $F=1/2$ ,  $AP=7/15$ ;

系统2查询1:  $P=2/4$ ,  $R=2/4$ ,  $F=1/2$ ,  $AP=3/8$ ; 系统2查询2:  $P=3/5$ ,  $R=3/3$ ,  $F=3/4$ ,  $AP=11/12$ ;

系统1的  $MacroP=2/5$ ,  $MacroR=7/12$ ,  $MacroF=17/36$ ,  **$MAP=29/60$** ,  $MicroP=4/10$ ,  
 $MicroR=4/7$ ,  $MicroF=8/17$

系统2的  $MacroP=11/20$ ,  $MacroR=3/4$ ,  $MacroF=5/8$ ,  **$MAP=31/48$** ,  $MicroP=5/9$ ,  
 $MicroR=5/7$ ,  $MicroF=5/8$

# 课堂提问

- 两个查询q1、q2的标准答案数目分别为100个和50个，某系统对q1检索出80个结果，其中正确数目为40，系统对q2检索出30个结果，其中正确数目为24，求MacroP/MacroR/MicroP/MicroR：

$$P1=40/80=0.5, R1=40/100=0.4$$

$$P2=24/30=0.8, R2=24/50=0.48$$

$$\text{MacroP}=(P1+P2)/2=0.65,$$

$$\text{MacroR}=(R1+R2)/2=0.44$$

$$\text{MicroP}=(40+24)/(80+30)=0.58$$

$$\text{MicroR}=(40+24)/(100+50)=0.43$$

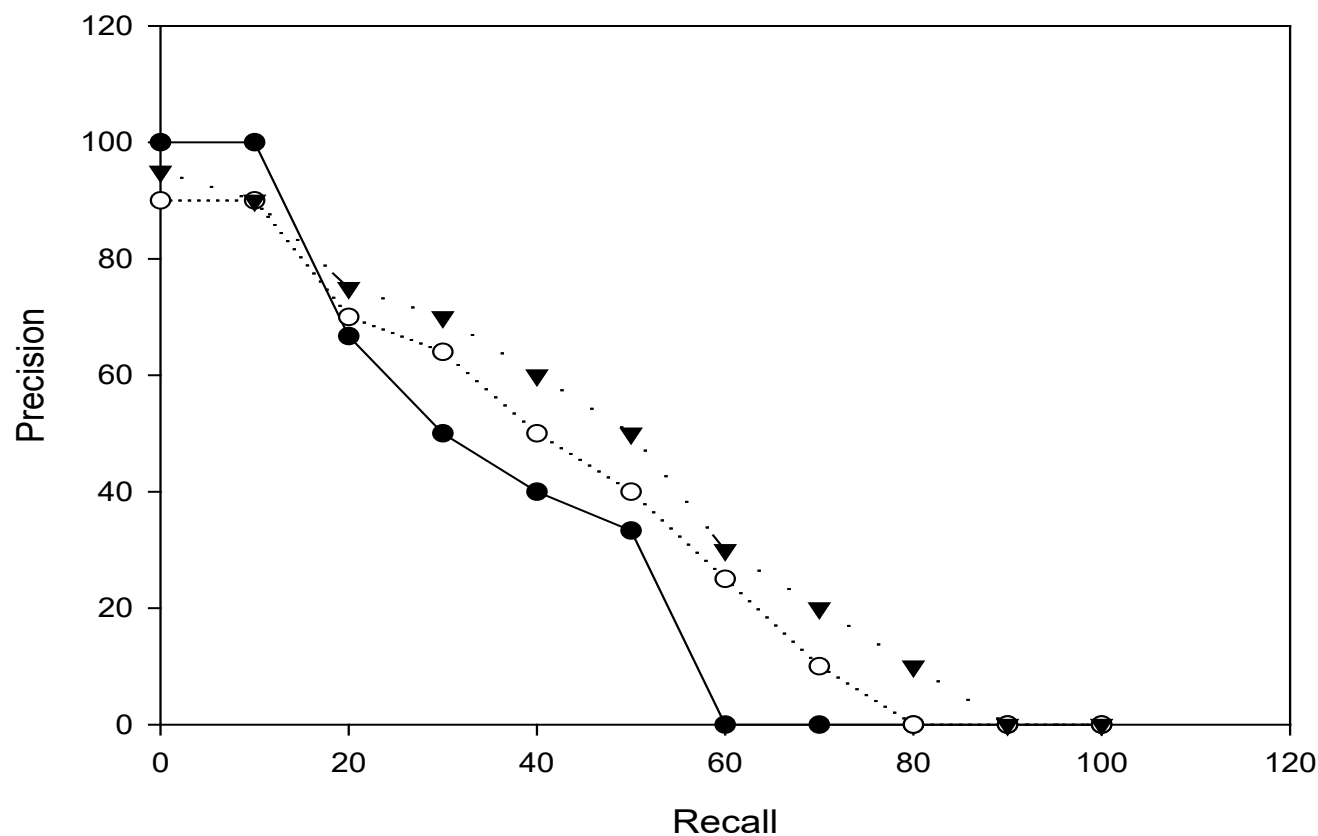
# 整个IR系统的P-R曲线

---

- 在每个召回率点上，对所有的查询在此点上的正确率进行算术平均，得到系统在该点上的正确率的平均值
- 两个检索系统可以通过P-R曲线进行比较，位置在上面的曲线代表的系统性能占优

# 几个IR系统的P-R曲线比较

几个系统的 P-R 曲线比较



# 关于相关性

- 相关性试图度量用户的满意程度，但是用户是否满意取决于很多因素
  - 用户工作量：用户是否容易构建查询、进行搜索以及浏览返回结果
  - 响应时间：输入到输入之间的等待时间
  - 结果呈现方式：用户方便浏览获得答案
  - 文档集覆盖度：文档集对相关文档的覆盖程度
- 搜索引擎往往还会考虑多样性(diversity)：结果的多样性，比如输入“苹果”，可以是公司、产品、操作系统、水果等等。

# 面向用户的评价指标

- 前面的指标都没有考虑用户因素。而相关不相关由用户判定。
- 假定用户已知的相关文档集合为 $U$ ，检索结果和 $U$ 的交集为 $R_u$ ，则可以定义覆盖率(Coverage)  
 $C=|R_u|/|U|$ ，表示系统找到的用户已知的相关文档比例。
- 假定检索结果中返回一些用户以前未知的相关文档 $R_k$ ，则可以定义出新率(Novelty Ratio)  
 $N=|R_k|/(|R_u|+|R_k|)$ ，表示系统返回的新相关文档的比例。



# 其他评价指标

- 不同的信息检索应用或者任务还会采用不同的评价指标
- 平均倒数排序MRR(Mean Reciprocal Rank):
  - 对于某些IR系统(如问答系统或主页发现系统), 只关心第一个标准答案返回的位置(Rank), 越前越好, 这个位置的倒数称为RR, 对问题集合求平均, 则得到MRR
  - 例子: 两个问题, 系统对第一个问题返回的标准答案的Rank是2, 对第二个问题返回的标准答案的Rank是4, 则系统的MRR为  $(1/2 + 1/4)/2 = 3/8$

# 近几年出现的新的评价指标

---

- GMAP
- NDCG

# GMAP

- GMAP(Geometric MAP): TREC2004 Robust 任务引进
- 先看一个例子

系统	Topic	AP	Increase	MAP
系统A	Topic 1	0.02	-	0.113
	Topic 2	0.03	-	
	Topic 3	0.29	-	
系统B	Topic 1	0.08	+300%	0.107
	Topic 2	0.04	+33.3%	
	Topic 3	0.20	-31%	

- 从MAP来看，系统A好于系统B，但是从每个查询来看，3个查询中有2个Topic B比A有提高，其中一个提高的幅度达到300%

# GMAP

- 几何平均值

$$GMAP = \sqrt[n]{\prod_{i=1}^n AP_i} = \exp\left(\frac{1}{n} \sum_{i=1}^n \ln AP_i\right)$$

- 上面那个例子  $GMAP_a=0.056$ ,  $GMAP_b=0.086$
- $GMAP_a < GMAP_b$
- GMAP和MAP各有利弊，可以配合使用，如果存在难Topic时，GMAP更能体现细微差别

# NDCG

---

- 每个文档不仅仅只有相关和不相关两种情况，而是有相关度级别，比如0,1,2,3。我们可以假设，对于返回结果：
  - 相关度级别越高的结果越多越好
  - 相关度级别越高的结果越靠前越好

\*Jarvelin, K. & Kekalainen, J. Cumulated Gain-based Evaluation of IR Techniques. *ACM Transactions on Information Systems*, **2002**, 20, 422-446

# NDCG

- 对某个查询 $q$ , Directed **G**ain

$$G' = \langle 3, 2, 3, 0, 0, 1, 2, 2, 3, 0, \dots \rangle.$$

- Cumulated Gain(**CG**) vector

$$CG[i] = \begin{cases} G[1], & \text{if } i = 1 \\ CG[i - 1] + G[i], & \text{otherwise.} \end{cases}$$

$$CG' = \langle 3, 5, 8, 8, 8, 9, 11, 13, 16, 16, \dots \rangle.$$

- Discounted CG** vector( ${}^b\log i$ 表示以 $b$ 为底对 $i$ 取对数)

$$DCG[i] = \begin{cases} CG[i], & \text{if } i < b \\ DCG[i - 1] + G[i]/{}^b\log i, & \text{if } i \geq b. \end{cases}$$

$$DCG' = \langle 3, 5, 6.89, 6.89, 6.89, 7.28, 7.99, 8.66, 9.61, 9.61, \dots \rangle.$$

# NDCG

- BV(Best Vector): 假定  $m$  个3,  $l$  个2,  $k$  个1, 其他都是0

$$BV[i] = \begin{cases} 3, & \text{if } i \leq m, \\ 2, & \text{if } m < i \leq m + l, \\ 1, & \text{if } m + l < i \leq m + l + k, \\ 0, & \text{otherwise.} \end{cases}$$

$$I' = \langle 3, 3, 3, 2, 2, 2, 1, 1, 1, 1, 0, 0, 0, \dots \rangle.$$

$$CG'_1 = \langle 3, 6, 9, 11, 13, 15, 16, 17, 18, 19, 19, 19, 19, \dots \rangle$$

$$DCG'_1 = \langle 3, 6, 7.89, 8.89, 9.75, 10.52, 10.88, 11.21, 11.53, 11.83, 11.83, 11.83, \dots \rangle.$$

# NDCG

---

- Normalized (D)CG

$$\text{norm-vect}(\mathbf{V}, \mathbf{I}) = \langle v_1/i_1, v_2/i_2, \dots, v_k/i_k \rangle.$$

$$\begin{aligned} \text{nCG}' &= \text{norm-vect}(\text{CG}', \text{CG}'_I) \\ &= \langle 1, 0.83, 0.89, 0.73, 0.62, 0.6, 0.69, 0.76, 0.89, 0.84, \dots \rangle. \end{aligned}$$

- $\text{N(D)CG}@k$ : 表示第 $k$ 个位置上的N(D)CG值



# 上述NDCG公式的另一种写法

$$CG[i] = \begin{cases} G[1], & \text{if } i = 1 \\ CG[i - 1] + G[i], & \text{otherwise.} \end{cases}$$

$$DCG[i] = \begin{cases} CG[i], & \text{if } i < b \\ DCG[i - 1] + G[i]/^b \log i, & \text{if } i \geq b. \end{cases}$$

- 上述公式中，令 $b=2$

$$DCG[k] = DCG[k - 1] + G[k] / \log_2 k$$

$$= DCG[k - 2] + \frac{G[k - 1]}{\log_2(k - 1)} + \frac{G[k]}{\log_2 k}$$

$$= DCG[1] + \frac{G[2]}{\log_2 2} + \dots + \frac{G[k]}{\log_2 k}$$

$$= G[1] + \frac{G[2]}{\log_2 2} + \dots + \frac{G[k]}{\log_2 k}$$

$$= G[1] + \sum_{m=2}^k \frac{G[m]}{\log_2 m}$$

$$DCG_I[k] = I[1] + \sum_{m=2}^k \frac{I[m]}{\log_2 m}$$

# 在查询集合上的平均

- 对于查询集合  $Q=\{q_1, q_2, \dots\}$  中的任一查询  $q_j$

$$NDCG(q_j, k) = \frac{DCG_j[k]}{DCG_{j,I}[k]} = \frac{G_j[1] + \sum_{m=2}^k \frac{G_j[m]}{\log_2 m}}{I_j[1] + \sum_{m=2}^k \frac{I_j[m]}{\log_2 m}}$$

- 上式中，令

$$Z_{j,k} = \frac{1}{I_j[1] + \sum_{m=2}^k \frac{I_j[m]}{\log_2 m}}$$

- 于是 
$$NDCG(Q, k) = \frac{1}{|Q|} \sum_{j=1}^{|Q|} Z_{j,k} (G_j[1] + \sum_{m=2}^k \frac{G_j[m]}{\log_2 m})$$

# NDCG

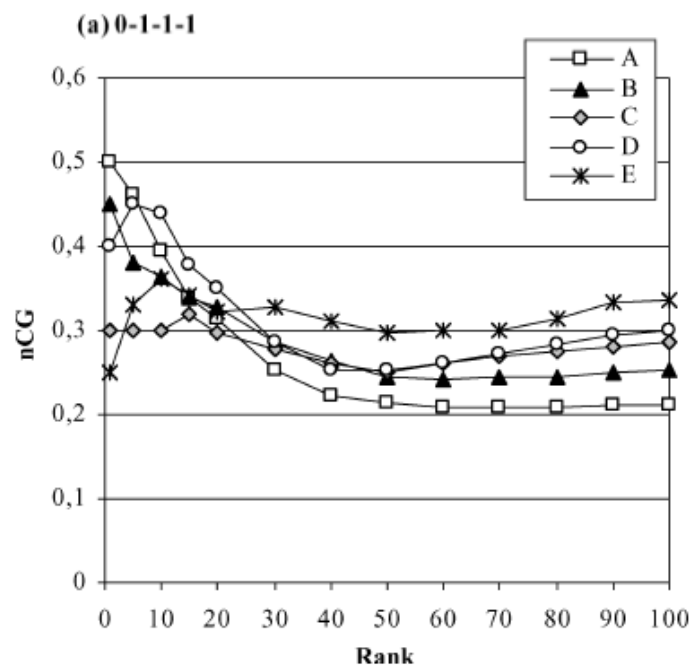


Fig. 3(a). Normalized cumulated gain (nCG) curves, binary weighting.

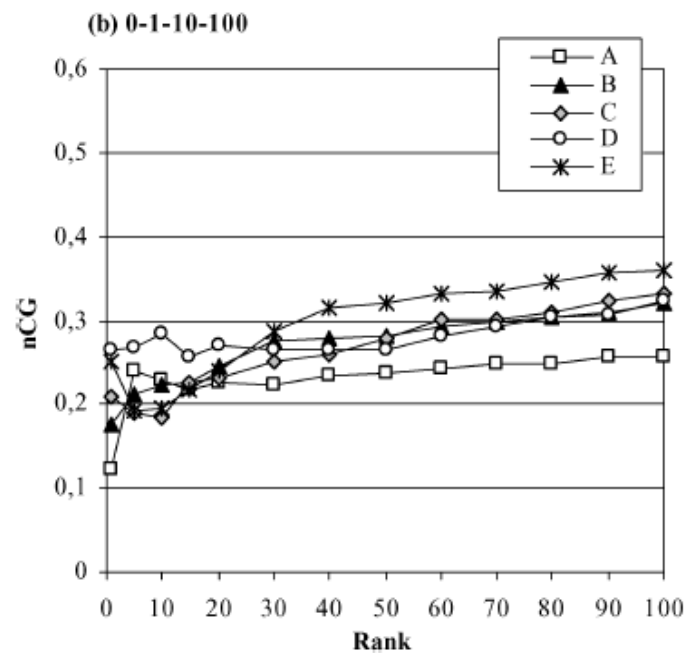


Fig. 3(b). Normalized cumulated gain (nCG) curves, nonbinary weighting.

# 另一种NDCG的计算方法

- 加大相关度本身的权重，原来是线性变化，现在是指数变化，相关度3、2、1 在计算时用 $2^3$ 、 $2^2$ 、 $2^1$

$$NDCG(Q, k) = \frac{1}{|Q|} \sum_{j=1}^{|Q|} Z_{j,k} (G_j[1] + \sum_{m=2}^k \frac{G_j[m]}{\log_2 m})$$



$$NDCG(Q, k) = \frac{1}{|Q|} \sum_{j=1}^{|Q|} Z_{j,k} (2^{G_j[1]} + \sum_{m=2}^k \frac{2^{G_j[m]}}{\log_2 m})$$

- 据说搜索引擎公司常用如下公式(书中给出的公式，和上述结果有点差异，请注意区分，其中 $G_j[m]=R(j,m)$ )

$$NDCG(Q, k) = \frac{1}{|Q|} \sum_{j=1}^{|Q|} Z_{j,k} \sum_{m=1}^k \frac{2^{R(j,m)} - 1}{\log(1 + m)}$$

# NDCG

---

- 优点：
  - 图形直观，易解释
  - 支持非二值的相关度定义，比P-R曲线更精确
  - 能够反映用户的行为特征(如：用户的持续性 persistence)
- 缺点：
  - 相关度的定义难以一致
  - 需要参数设定

\*Ruihua Song, Evaluation in Information Retrieval, 中科院研究生院微软系列讲座,  
<http://tjluo.gucas.ac.cn/sites/wism2006/PPT/Forms/AllItems.aspx>

# 关于评价方面的研究

---

- 现有评价体系远没有达到完美程度
  - 对评价的评价研究
  - 指标的相关属性(公正性、敏感性)的研究
  - 新的指标的提出(新特点、新领域)
  - 指标的计算(比如Pooling方法中如何降低人工代价? 查询集或文档集合发生变化怎么办? )

推荐阅读: Ben Carterette, James Allan and Ramesh Sitaraman, Minimal Test Collections for Retrieval Evaluation, SIGIR06, best paper & best student paper.

# 提纲

- ① 上一讲回顾
- ② 有关检索评价
- ③ 评价指标
- ④ 相关评测
- ⑤ 结果摘要

# TREC 概况

---

- The Text REtrieval Conference, TREC, <http://trec.nist.gov>
- 由美国NIST(the National Institute of Standards and Technology)和DARPA(the Defense Advanced Research Projects Agency)联合举办
- 1992年举办第一届会议，每年11月举行，至2019年已有28届，可以看成信息检索领域的“奥运会”



# TREC的目标

---

- 总目标：支持在信息检索领域的基础研究，提供对大规模文本检索方法的评估办法
- 1. 鼓励对基于大测试集合的信息检索方法的研究
- 2. 提供一个可以用来交流研究思想的论坛，增进工业界、学术界和政府部门之间的互相了解；
- 3. 示范信息检索理论在解决实际问题方面的重大进步，提高信息检索技术从理论走向商业应用的速度；
- 4. 为工业界和学术界提高评估技术的可用性，并开发新的更为适用的评估技术。

# TREC的运行方式(1)

---

- TREC由一个程序委员会管理。这个委员会包括来自政府、工业界和学术界的代表。
- TREC以年度为周期运行。过程为：确定任务→参加者报名→参加者运行任务→返回运行结果→结果评估→大会交流
- 一开始仅仅面向文本，后来逐渐加入语音、图像、视频方面的评测

# TREC的运行方式(2)

---

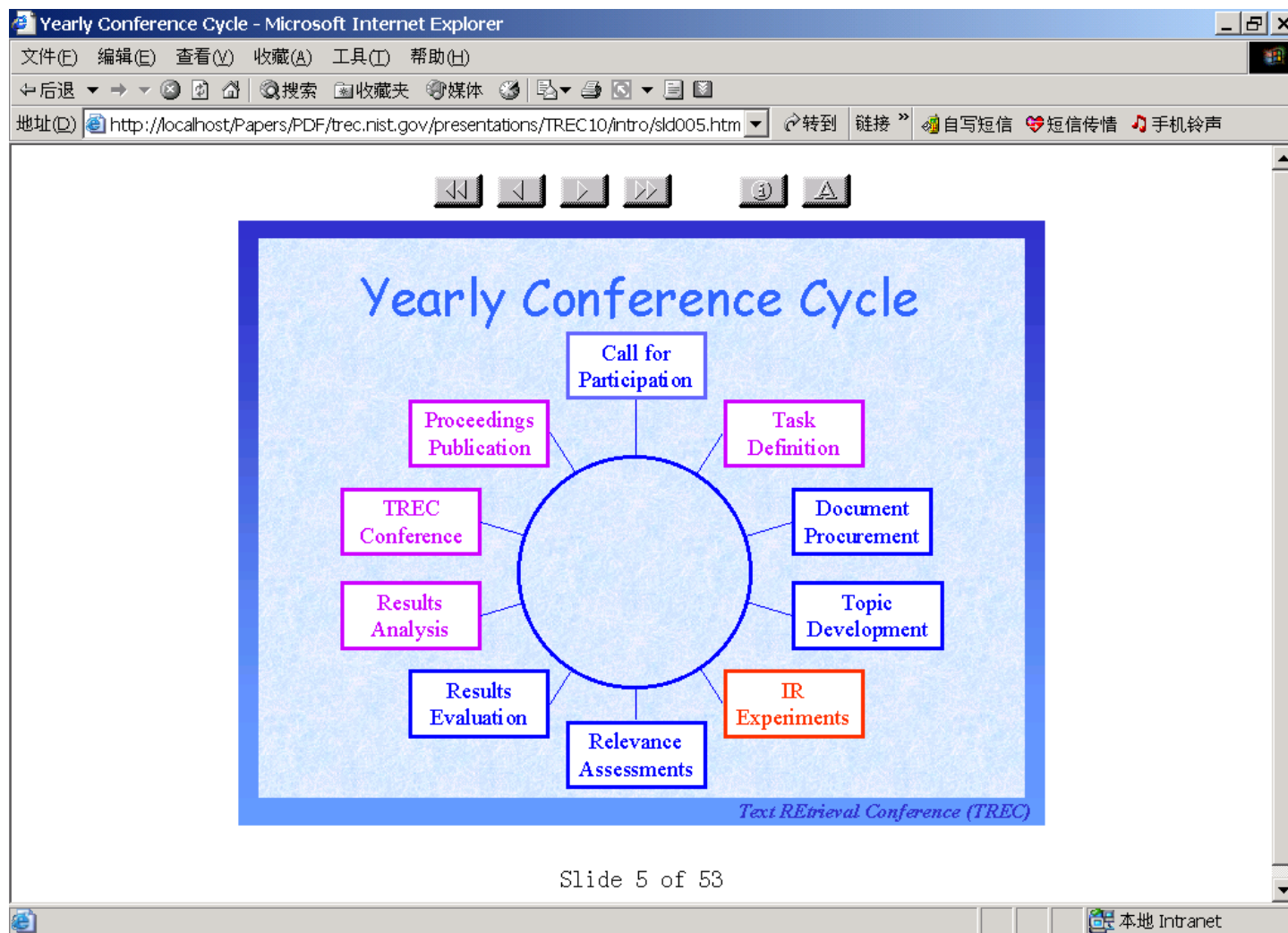
- 确定任务：NIST提供测试数据和测试问题
- 报名：参加者根据自己的兴趣选择任务
- 运行任务：参加者用自己的检索系统运行测试问题，给出结果
- 返回结果：参加者向NIST返回他们的运行结果，以便评估

# TREC的运行方式(3)

---

- 结果评估：NIST使用一套固定的方法和软件对参加者的运行结果给出评测结果
- 大会交流：每年的11月召开会议，由当年的参加者们交流彼此的经验

# TREC的运行方式(4)



# 测试数据和测试软件

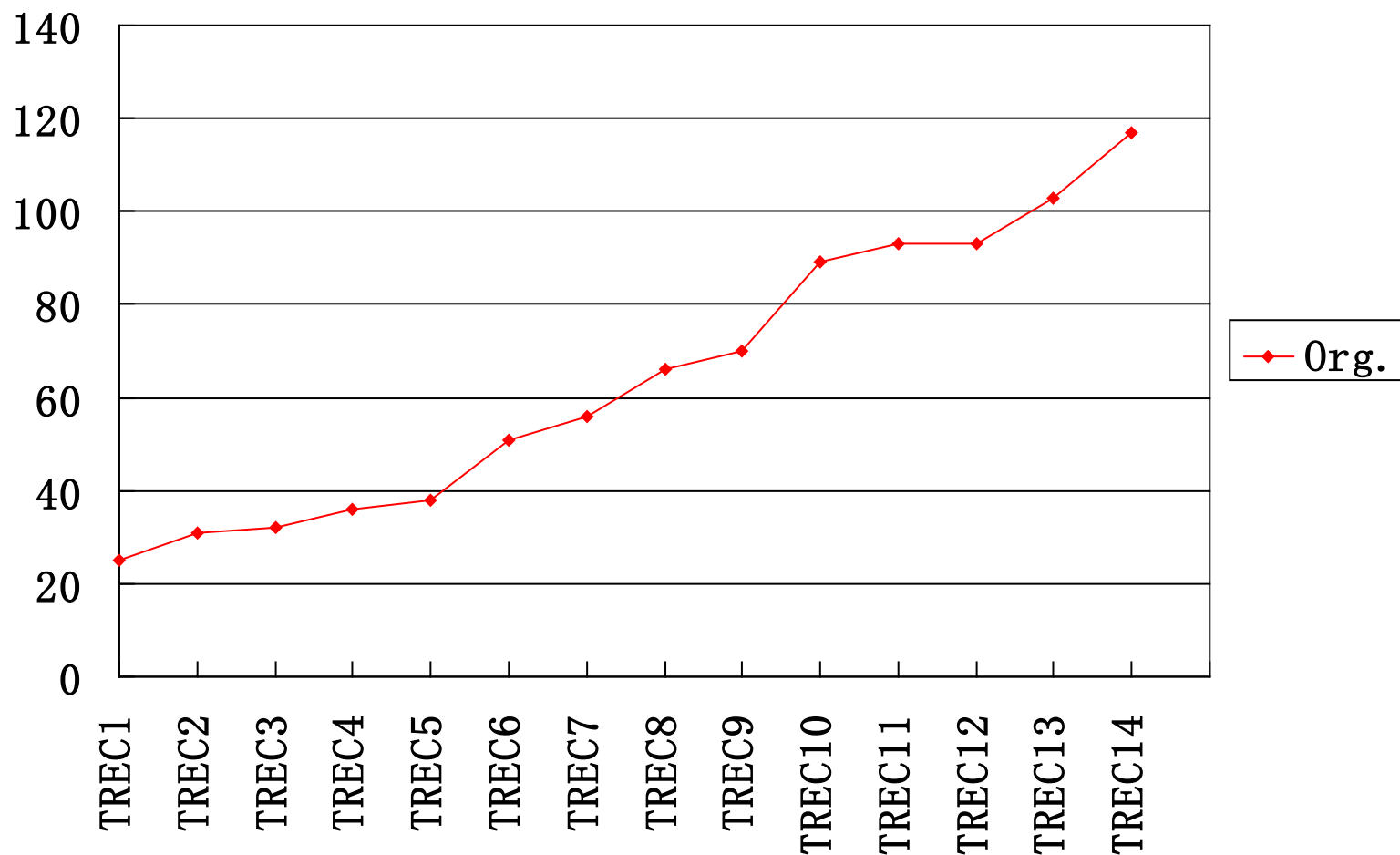
---

- 由LDC([Linguistic Data Consortium](#))或者其他单位免费提供，但有些数据需要缴纳费用，一般都必须签订协议
- 每年使用的数据可以是新的，也可以是上一年度已经使用过的
- TREC使用的评估软件是开放的，任何组织和个人都可以用它对自己的系统进行评测

# TREC任务情况

TREC1 (92)	25	Ad hoc/Routing
TREC2	31	Ad hoc/Routing
TREC3	32	Ad hoc/Routing
TREC4	36	Spanish/Interactive/Database Merging/Confusion/Filtering
TREC5	38	Spanish/Interactive/DatabaseMerging/Confusion/Filtering/NLP
TREC6	51	Chinese/Interactive/Filtering/NLP/CLIR/Highprecision/SDR/VLC
TREC7	56	CLIR/High Precision/Interactive/Query/SDR/VLC
TREC8	66	CLIR/Filtering/Interactive/QA/Query/SDR/Web
TREC9	70	QA/CLIR(E-C)/Web/Filtering/Interactive/Query/SDR
TREC10	89	QA/CLIR/Web/Filtering/Interactive/Video
TREC11 (02)	93	QA/CLIR/Web/Filtering/Interactive/Video/Novelty/
TREC12 (03)	93	QA/Web/Novelty/HARD/Robust/Genomics/ →TRECVID单独组织
TREC13 (04)	103	QA/Web/Novelty/HARD/Robust/Genomics/Terabyte
TREC14 (05)	117	QA/HARD/Robust/Enterprise/Genomics/Terabyte/SPAM
TREC15 (06)	n/a	QA/Legal/Enterprise/Genomics/Terabyte/SPAM/Blog
TREC16 (07)	n/a	QA/Legal/Enterprise/Genomics/Terabyte/SPAM/Blog/Million Query

# 历届TREC参加单位数示意图





# 参加过TREC的部分单位

Corp.	University	Asian Organization
IBM	MIT	Singapore U. (KRDL)
AT&T	CMU	KAIST
Microsoft	Cambridge U.	Tinghua U. (大陆的清华) TREC11
Sun	Cornell U.	<a href="#">Tsinghua U.</a> (Taiwan) TREC7
Apple	Maryland U.	Taiwan U. TREC8&9&10
Fujitsu	Massachusetts U.	Hongkong Chinese U. TREC9
NEC	New Mexico State U.	Microsoft Research China TREC9&10
XEROX	California Berkeley U.	Fudan U. TREC9&10&11(复旦)
RICOH	Montreal U.	ICT TREC10&11(中科院计算所)
CLRITECH	Johns Hopkins U.	HIT TREC10(哈工大)
NTT	Rutgers U.	北大、软件所、自动化所等
Oracle	Pennsylvania U.	还有更多的大陆队伍逐渐加入.....

# TREC中名词定义

---

- Track
  - TREC的每个子任务，QA、Filtering、Web、Blog等
- Topic
  - 预先确定的问题，用来向检索系统提问
  - topic→query (自动或者手工)
  - Question (QA)
- Document
  - 包括训练集和测试集合 (TIPSTER&TREC CDs、WT2G、WT10G、GOV2)
- Relevance Judgments
  - 相关性评估，人工或者半自动

# Topic的一般结构

---

- **Title:** 标题，通常由几个单词构成，非常简短
- **Description:** 描述，一句话，比Title详细，包含了Title的所有单词
- **Narrative:** 详述，更详细地描述了哪些文档是相关的

# Topic示例

---

**<num>** Number: 351

**<title>** Falkland petroleum exploration

**<desc>** Description:

What information is available on petroleum exploration in the South Atlantic near the Falkland Islands?

**<narr>** Narrative:

Any document discussing petroleum exploration in the South Atlantic near the Falkland Islands is considered relevant. Documents discussing petroleum exploration in continental South America are not relevant.

# 使用Topic的方式

---

- 按照会议要求，可以利用Topic文本中的部分或者全部字段，构造适当的查询条件
- 可以使用任何方式构造查询条件，这包括手工的和自动的两大类。但提交查询结果时要注明产生方式。

# 评测方法

---

- 基于无序集合的评测：返回结果无顺序
  - Set Precision/Set Recall
- 基于有序集合的评测：
  - $P@n$ /Average Precision/Reciprocal Rank
- 其他评测方法
  - Filtering Utility

# 相关性评估过程(1)

---

- (Ad hoc任务)Pooling方法：对于每一个topic，NIST从参加者取得的结果中挑选中一部分运行结果，从每个运行结果中取头N个文档，然后用这些文档构成一个文档池，使用人工方式对这些文档进行判断。相关性判断是二值的：相关或不相关。没有进行判断的文档被认为是不相关的。

# 相关性评估过程(2)

---

- NIST使用trec\_eval软件包对所有参加者的运行结果进行评估，给出大量参数化的评测结果（主要是precision和recall）。根据这些评测数据，参加者可以比较彼此的系统性能。
- 其他track也有相应的公开评测工具



# 其他评测会议

# TRECVID (TREC VIdeo)

---

- TRECVID: 2003年从TREC中分出来的有关Video检索方面的评测, 之前是TREC中的Video track任务。

# MUC (Message Understanding Conference)

- 美国DARPA组织的有关信息抽取(IE, Information Extraction)的评测会议，起于1991年，1997年为最后一届(后来演变为ACE评测)，后两届加入了命名实体(Name Entity)识别和共指(Co-reference)消解

Conference	Year	Text Source	Topic (Domain)
MUC-1	1987	Mil. reports	Fleet Operations
MUC-2	1989	Mil. reports	Fleet Operations
MUC-3	1991	News reports	Terrorist activities in Latin America
MUC-4	1992	News reports	Terrorist activities in Latin America
MUC-5	1993	News reports	Corporate Joint Ventures, Microelectronic production
MUC-6	1995	News reports	Negotiation of Labor Disputes and Corporate management Succession
MUC-7	1997	News reports	Airplane crashes, and Rocket/Missile Launches

# ACE(Automatic Content Extraction)

---

- 美国NIST组织，主要面向新闻领域的文本，抽取其中的实体、关系和事件。2000年开始，每年1届(2006年停办1次)，目前是进行了八届。ACE是以对象(Object)为单位进行提取，而MUC是以词语为单位进行提取。

# DUC(Document Understanding Conference)

---

- 2001年开始NIST等开始组织的面向文档摘要(Summarization)的评测会议，评测的任务有单文档摘要和多文档摘要，通用摘要和面向查询(query-biased)的摘要，目前已经进行到第八届

# 其他评测

- NTCIR(NII Test Collection for IR Systems)：日本国立情报学研究所组织的关于亚洲语言相关的IR评测，1998年11月开始-1999年9月为第一届
- CLEF：有关欧洲语言相关的IR评测(跨语言)
- TAC(Text analysis Conference)：将DUC任务和TREC中的QA任务合并，自2008年开始举办的一个新会议。
- INEX(Initiative for the evaluation of XML retrieval)：有关XML检索的一个评测，起于2002年，DELOS Network of Excellence for Digital Libraries和IEEE CS组织。
- 国内863评测、北大天网评测、中文信息学会的倾向性分析评测等等

# 用户判定/标注的有效性

---

- 只有在用户的评定一致时，相关性判定的结果才可用
- 如果结果不一致，那么不存在标准答案无法重现实验结果
- 如何度量不同判定人之间的一致性？
- → Kappa 指标

# Kappa (1)

---

- Kappa是度量判定间一致性的指标
- 为类别性判断结果(判定的结果是类别型)所设计的指标
- 对随机一致性的修正
- $P(A)$  = 观察到的一致性判断比例
- $P(E)$  = 随机情况下所期望的一致性判断比例

$$\kappa = \frac{P(A) - P(E)}{1 - P(E)}$$



# Kappa (2)

---

- $kappa$  在  $[2/3, 1.0]$  时，判定结果是可以接受的
- 如果  $kappa$  值比较小，那么需要对判定方法进行重新设计

# 计算kappa统计量

		Judge 2 Relevance		
		Yes	No	Total
Judge 1 Relevance	Yes	300	20	320
	No	10	70	80
	Total	310	90	400

- 观察到的一致性判断比率:

$$P(A) = (300 + 70)/400 = 370/400 = 0.925$$

- 边缘统计量:

$$P(\text{nonrelevant}) = (80 + 90)/(400 + 400) = 170/800 = 0.2125$$

$$P(\text{relevant}) = (320 + 310)/(400 + 400) = 630/800 = 0.7878$$

- 两人的随机一致性比率:

$$P(E) = P(\text{nonrelevant})^2 + P(\text{relevant})^2 = 0.2125^2 + 0.7878^2 = 0.665$$

- Kappa统计量:

$$\kappa = (P(A) - P(E))/(1 - P(E)) = (0.925 - 0.665)/(1 - 0.665) = 0.776$$

# TREC中判定的一致性情况

信息需求	判断文档数	不一致数目
51	211	6
62	400	157
67	400	68
95	400	110
127	400	106

# 不一致性带来的影响

---

- 上述的不一致性很严重。这是否意味着信息检索的实验结果没有意义？
- 不是的。
- 不一致性会对指标的绝对数值有很大影响
- 事实上对系统之间的相对排序没有影响
- 比如，我们想知道A算法是否好于B算法
- 信息检索实验会给出一个可靠的答案，即使判定人员之间的不一致性可能很大

# 大型搜索引擎的评价

- Web下召回率难以计算
- 搜索引擎常使用top  $k$ 的正确率来度量, 比如,  $k = 10 \dots$
- $\dots$  或者使用一个考虑返回结果所在位置的指标, 比如正确答案在第一个返回会比第十个返回的系统给予更大的指标
- 搜索引擎也往往使用非相关度指标
  - 比如: 第一个结果的点击率
  - 仅仅基于单个点击使得该指标不太可靠 (比如你可能被检索结果的摘要所误导, 等点进去一看, 实际上是不相关的)  $\dots$
  - 当然, 如果考虑点击历史的整体情况会相当可靠
  - 比如: 一些基于用户行为的指标
  - 比如: A/B 测试

# A/B 测试

---

- 目标：测试某个独立的创新点
- 先决条件：大型的搜索引擎已经在线上运行
- 很多用户使用老系统
- 将一小部分(如1%)流量导向包含了创新点的新系统
- 对新旧系统进行自动评价，并得到某个评价指标，比如第一个结果的点击率
- 于是，可以通过新旧系统的指标对比来判断创新点的效果
- 这也可能是大型搜索引擎最信赖的方法

# 上述相关性定义的补充

- 上述相关性定义是针对独立的查询-文档对
- 另一种定义：边缘相关性(marginal relevance)
- 边缘相关性指的是结果列表中位置 $k$ 上的文档相对于其前面的文档  $d_1 \dots d_{k-1}$ 中包含的信息之外所带来的额外信息。同一篇文档，后面再次出现不能带来更多的信息。
- 课堂练习
  - 为什么边缘相关性更能表示用户的真实满意度？
  - 给出一个P、R类相关性指标不能反映用户满意度而边缘相关性却能否反映用户满意度的例子。
  - 在实际系统中，使用边缘相关性的难点在哪？

# 关于评价的可信度

- 在多组数据集上评价，得到相似的结果
- 进行统计显著性检测(statistical significance test)
  - 配对t检验
  - 符号检验
  - Wilcoxon符号秩检验

查询	系统一的 AP (X)	系统二的 AP (Y)	X-Y
1	0.0273	0.0323	-0.005
2	0.5725	0.5796	-0.0071
3	0.1388	0.1772	-0.0384
4	0.1196	0.1066	0.013
5	0.0015	0.0033	-0.0018
6	0.1069	0.1093	-0.0024
7	0.2127	0.2311	-0.0184
8	0.3617	0.4414	-0.0797
9	0.0005	0.001	-0.0005
10	0.1636	0.1426	0.021
11	0.3618	0.4206	-0.0588
12	0.7412	0.7412	0
13	0.6814	0.7866	-0.1052
14	0.0019	0.0023	-0.0004
15	0.0362	0.0113	0.0249
均值	0.2352	0.2524	-0.0172



# 提纲

- ① 上一讲回顾
- ② 有关检索评价
- ③ 评价指标
- ④ 相关评测
- ⑤ 结果摘要

# 结果的呈现

---

- 最常见的就是列表方式，也称为 “10 blue links”
- 怎样描述该列表中的每篇文档？
- 该描述很关键
  - 用户往往根据该描述来判断结果的相关性
  - 而不需要按次序点击所有文档

# 文档描述方式

- 最常见的方式：文档标题、url以及一些元数据
- . . . 以及一个摘要(snippet)

[PDF] [Minimal Test Collections for Retrieval Evaluation - Center for ...](http://ciir.cs.umass.edu/~carteret/sigir06.pdf)  
[ciir.cs.umass.edu/~carteret/sigir06.pdf](http://ciir.cs.umass.edu/~carteret/sigir06.pdf) ▾ 翻译此页

作者：B Carterette - 2006 - 被引用次数：160 - 相关文章

2006年8月11日 - Minimal Test Collections for Retrieval Evaluation. Ben Carterette carteret@cs.umass.edu. James Allan allan@cs.umass.edu. Ramesh ...

- 如何计算摘要？

# 摘要

---

- 两种基本类型：(i) 静态 (ii) 动态
- 不论输入什么查询，文档的静态摘要都是不变的
- 而动态摘要依赖于查询，它试图解释当前文档返回的原因

# 静态摘要

- 一般系统中静态摘要是文档的一个子集
- 最简单的启发式方法：返回文档的前50个左右的单词作为摘要
- 更复杂的方法：从文档中返回一些重要句子组成摘要
  - 可以采用简单的NLP启发式方法来对每个句子打分
  - 将得分较高的句子组成摘要
  - 也可以采用机器学习方法，参考第13章
- 最复杂的方法：通过复杂的NLP方法合成或者生成摘要
  - 对大部分IR应用来说，最复杂的方法还不够成熟

# 动态摘要

---

- 给出一个或者多个“窗口”内的结果(snippet), 这些窗口包含了查询词项的多次出现
- 出现查询短语的snippet优先
- 在一个小窗口内出现查询词项的snippet优先
- 最终将所有snippet都显示出来作为摘要

# 一个动态摘要的例子

查询: “new guinea economic development” Snippets (加黑标识) that were extracted from a document: . . . **In recent years, Papua New Guinea has faced severe economic difficulties and** economic growth has slowed, partly as a result of weak governance and civil war, and partly as a result of external factors such as the Bougainville civil war which led to the closure in 1989 of the Panguna mine (at that time the most important foreign exchange earner and contributor to Government finances), the Asian financial crisis, a decline in the prices of gold and copper, and a fall in the production of oil. **PNG’s economic development record over the past few years is evidence that** governance issues underly many of the country’s problems. Good governance, which may be defined as the transparent and accountable management of human, natural, economic and financial resources for the purposes of equitable and sustainable development, flows from proper public sector management, efficient fiscal and accounting mechanisms, and a willingness to make service delivery a priority in practice. . . .

# Google中的动态摘要



# 动态摘要的生成

---

- 需要对文档进行缓存
- 通过位置索引会知道查询词项在文档中的出现位置
- 文档的缓存版本可能会过时
- 不缓存非常长的文档，对这些文档只需要缓存其一个短前缀文档

# 动态摘要

- 搜索结果页面的空间是有限的，snippet必须要短
- 但是另一方面要使snippet有意义，它们又要足够长
- 通过Snippet应该能够判断文档的相关性
- 理想情况：语言上良构的snippet
- 理想情况： snippet就能回答查询而不需要继续浏览文档
- 动态摘要对于用户满意度相当重要
  - 用户可以快速浏览这些snippet从而确定是否需要点击
  - 很多情况下，我们根本不需要点击就可以获得答案，从而可以节省时间

# 本讲小结

---

- 信息检索的评价方法
  - 不考虑序检索评价指标(即基于集合): P、R、F
  - 考虑序的评价指标: P/R曲线、MAP、NDCG
- 信息检索评测语料及会议
- 检索结果的摘要

# 参考资料

---

- 《信息检索导论》 第8章
- TREC主页: <http://trec.nist.gov>
- *F*-measure: Keith van Rijsbergen
- 更多有关 A/B 测试的文章
- Too much A/B testing at Google?
- Tombros & Sanderson 1998: 动态摘要的最早的几篇文章之一
- Google VP of Engineering on search quality evaluation at Google

# 课后练习

---

- 有待补充