检索评价

目录

- 有关检索评价
- 无序检索结果的评价
- 有序检索结果的评价
- 结果摘要

关于评价

- 评价无处不在，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

为什么要评估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。

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

如何评价效果？

- 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 – 今 (上亿篇文档)，信息检索的“奥运会”

评价任务的例子

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	d3	d9	
系统2，查询2	d1	d2	d4	d13	

- 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？

评价的几部分

- 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

目录

- 有关检索评价
- 无序检索结果的评价
- 有序检索结果的评价
- 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

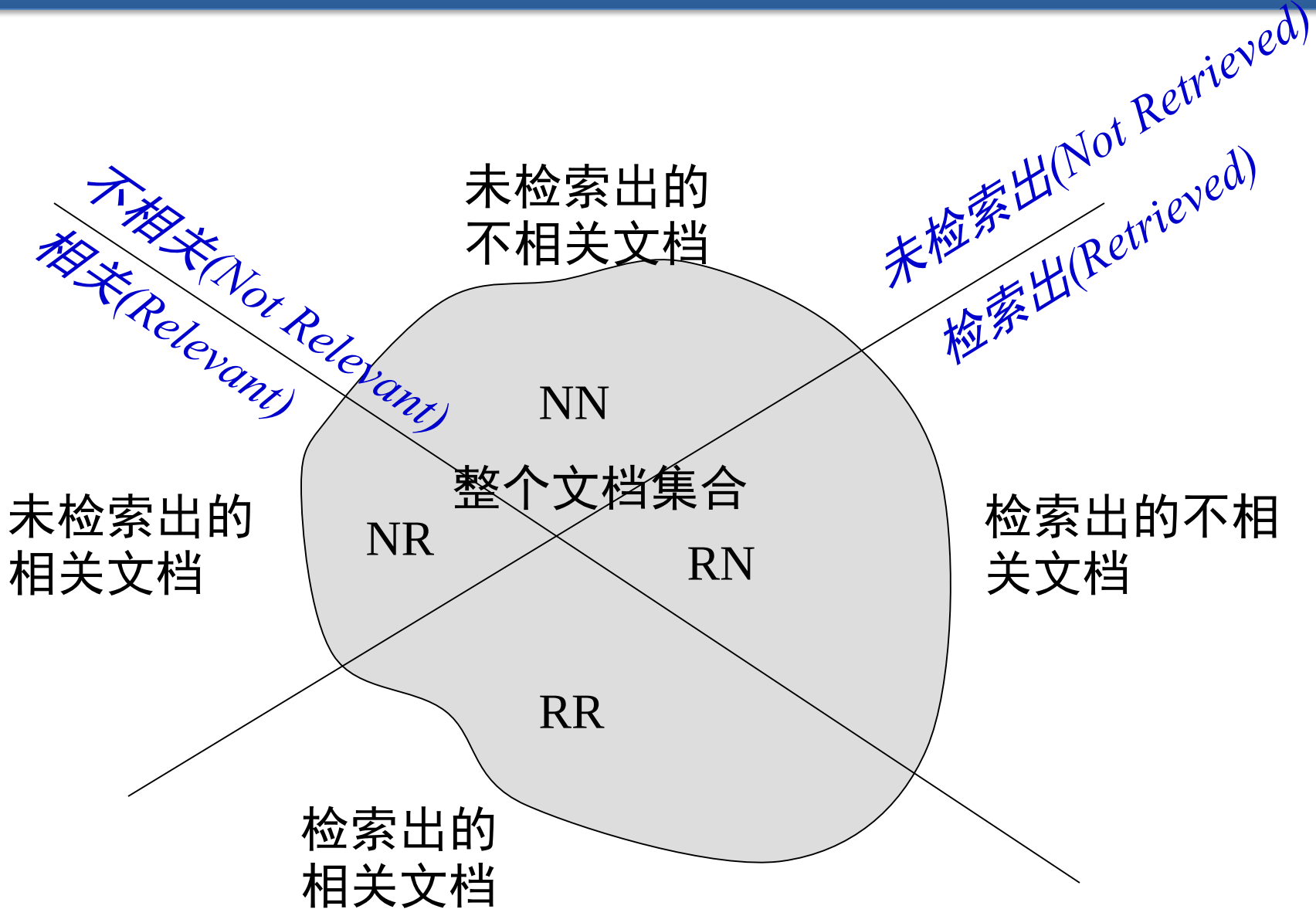
- 对多个查询进行评估的指标
  - 在多个查询上检索系统的得分

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	

对于查询1的标准答案集合 {d3,d4,d6,d9}

整个文档集合的划分


不相关(Not Relevant)
相关(Relevant)
未检索出(Not Retrieved)
检索出(Retrieved)
未检索出的
不相关文档
NN
整个文档集合
未检索出的
相关文档
NR
RN
检索出的不相
关文档
RR
检索出的
相关文档

评价指标

- **召回率(Recall):** 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∈ [0,1]

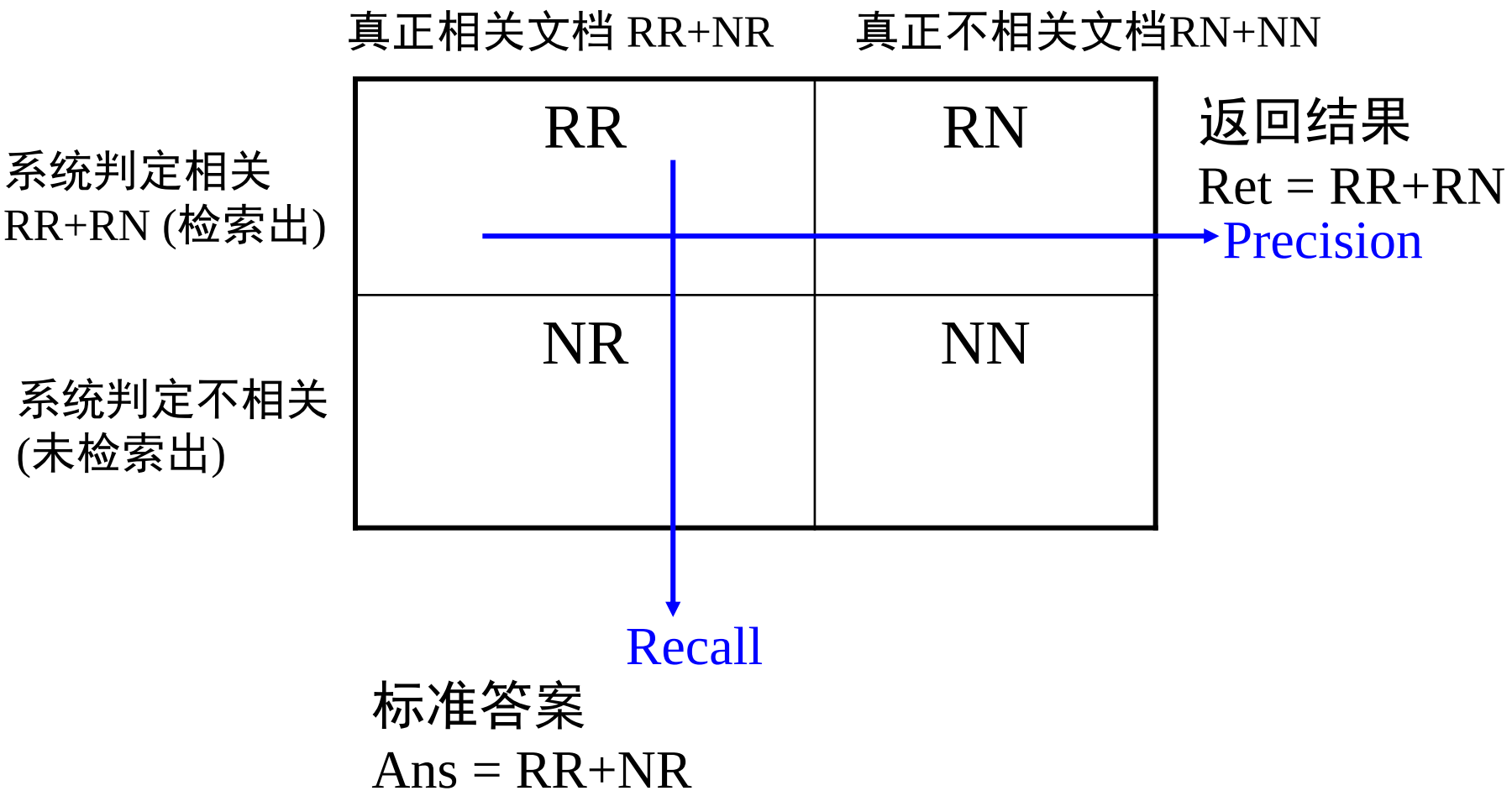
$$R = \frac{RR}{RR + NR}$$

- **正确率(Precision):**返回的结果中真正相关结果的比率，也称为**查准率**， P∈ [0,1]

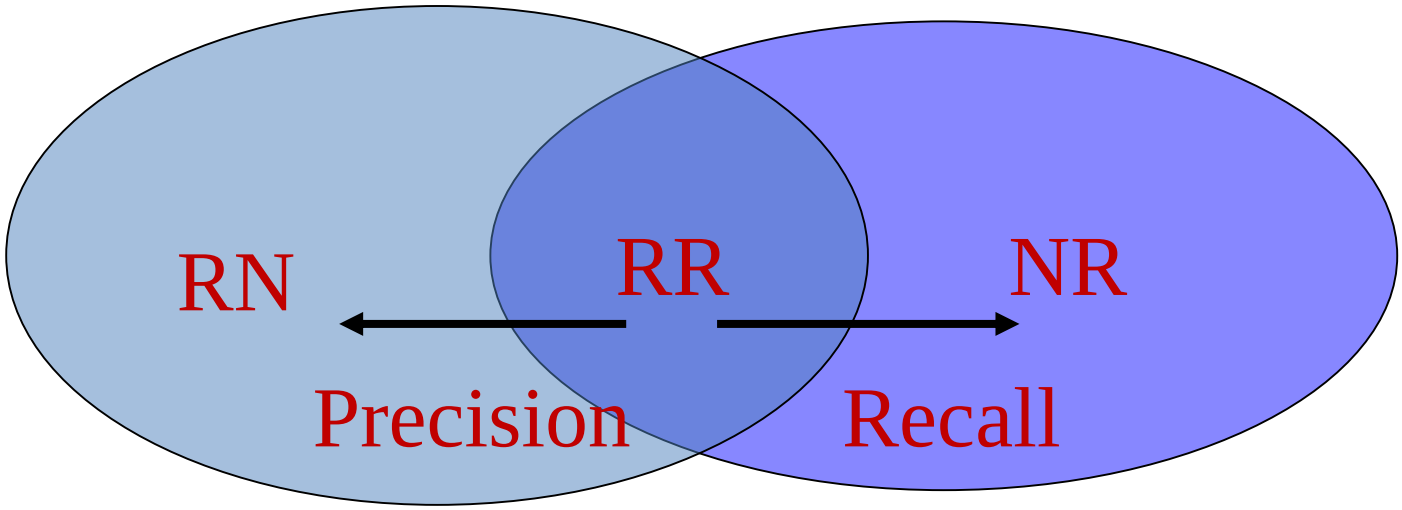
$$P = \frac{RR}{RR + RN}$$

- 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RN+NN
RR
RN
NR
NN
系统判定相关
RR+RN (检索出)
系统判定不相关
(未检索出)
返回结果
Ret = RR+RN
Precision
Recall
标准答案
Ans = RR+NR

基于集合的图表示



返回结果Ret　　标准答案Ans

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，查准率2/5，召回率2/4

对于系统2，查询1，查准率2/4，召回率2/4

查准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
- ……

关于查准率和召回率的讨论

- “宁可错杀一千，不可放过一人”➔偏重召回率，忽视查准率。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
- 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。

P/R指标的方差

- 对于一个测试文档集来说，某些信息需求上效果很差(比如，在 $R = 0.1$ 点上 $P = 0.2$)，但是在一些其它需求上又相当好 (如在$R = 0.1$ 点上$P = 0.95$)
- 实际上，同一系统在不同查询上的结果差异往往高于不同系统在同一查询上的结果
- 也就是说，存在容易的信息需求和难的信息需求

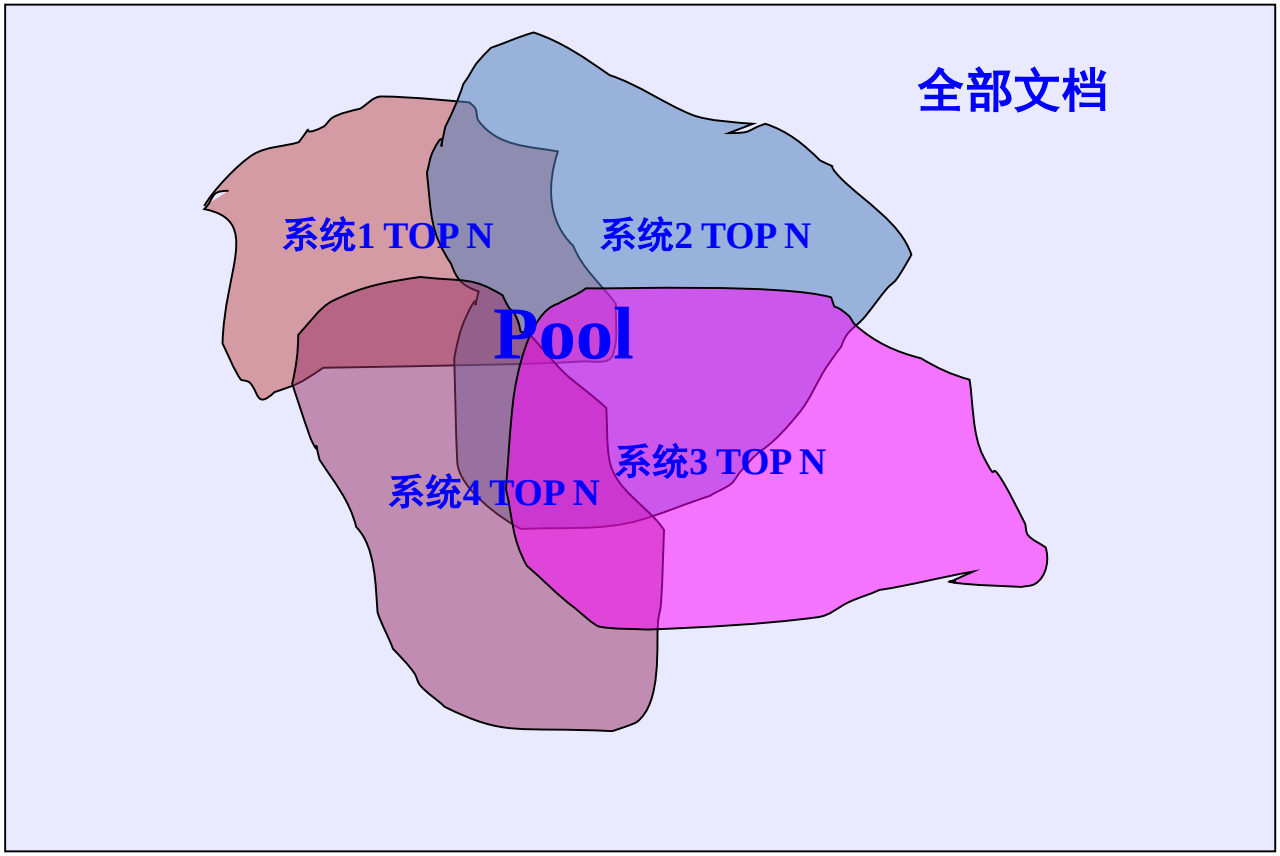
查准率和召回率的问题

- 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？
  - 解决方法：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**

关于召回率的计算

- 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，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 *N*个结果组成的集合进行人工标注，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



使用查准率/查全率的问题

- 需要在大规模的文档集合和查询集合上进行计算
- 需要人工对返回的文档进行评价
  - 由于人的主观因素，人工评价往往不可靠
- 评价是二值的
  - 无法体现细微的差别
- 文档集合和数据来源不同，结果也不同，有严重的偏差
  - 评价结果只适用于某个范围，很难引申到其他的范围

一个综合评价准则：$F = P$和R融合

- F值(F-measure)：召回率R和查准率P的加权调和平均值，
  - if P=0 or R=0, then F=0,
  - else：

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

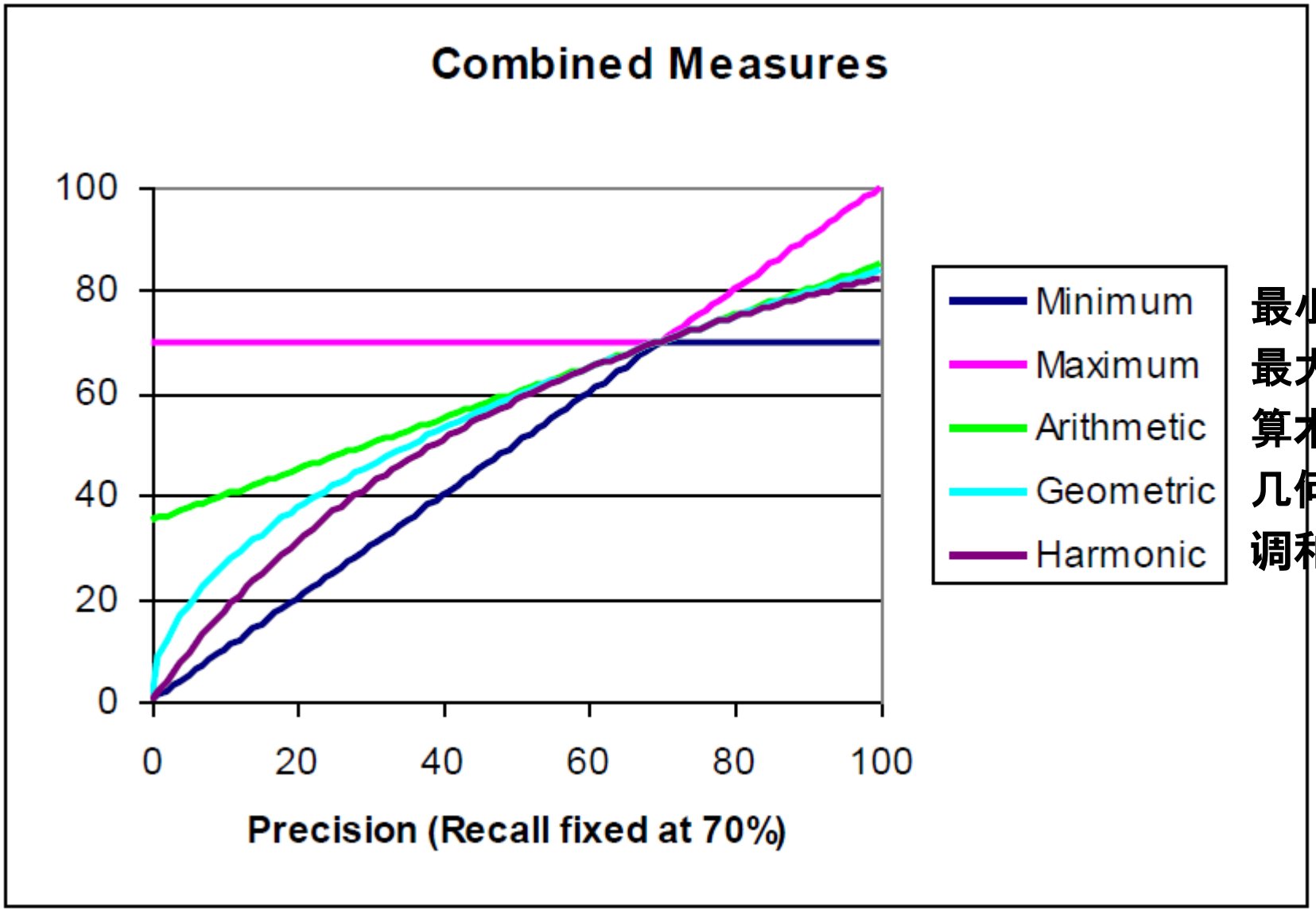
- F_β：表示召回率的重要程度是查准率的β(>=0)倍
  - β>1更重视召回率，β<1更重视查准率
  - 一般取等权重

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

为什么使用调和平均计算F值

- 调和平均比较“保守”
  - 调和平均小于算术平均和几何平均
  - 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%，这有些过高。
- 做法：不管是P还是R，如有一个偏低，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

$F_{\beta=1}$ 和其他平均数的比较



最小
最大
算术平均
几何平均
调和平均

精确率(Accuracy)

- 精确率是所有判定中正确的比率，即被正确判定(相关→相关，不相关→不相关)的文档占总文档的百分比
  - accuracy = (RR+NN)/(RN + RR + NR + NN)

- 为什么通常使用*P*、*R*、*F*而不使用精确率？
- 信息检索当中精确率为什么不可用？

填空题 3分

设置

	相关	不相关
返回	18	2
未返回	82	1,000,000,000

计算P= [填空1] 、R= [填空2] 、F1= [填空3] 。(请用小数)

正常使用填空题需3.0以上版本雨课堂

作答

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

- P=18/20， R=18/100，

A=(18+1000000000)/(18+2+82+1000000000)

- 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

- 由于和查询相关的文档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

- 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

目录

- 有关检索评价
- 无序检索结果的评价
- 有序检索结果的评价
- 结果摘要

评价排序后的结果

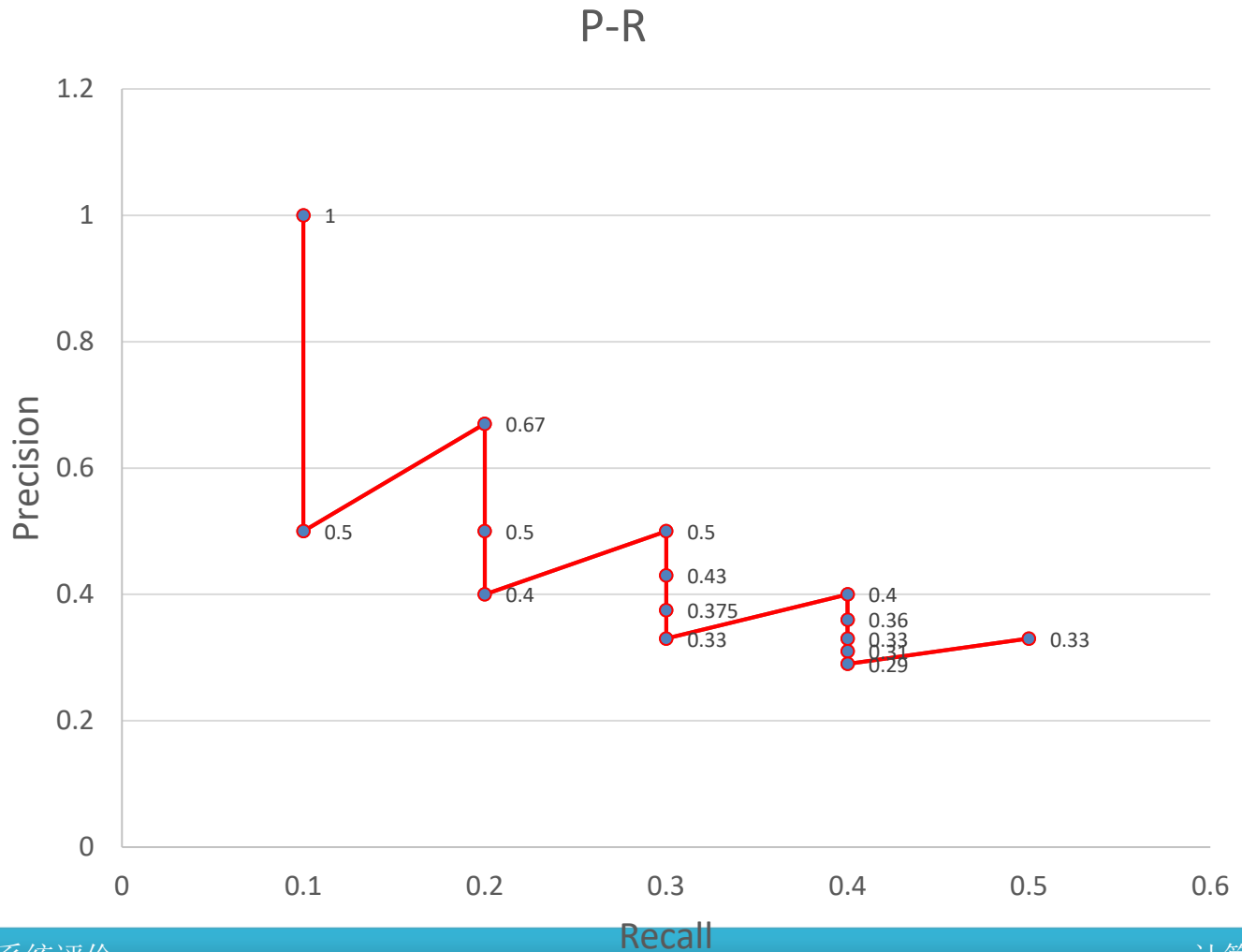
- P、R、F值都是基于无序集合的评价方法。
  - →如果搜索引擎输出为有序的检索结果时，需要扩展。
- 对于一个特定检索词的有序检索结果
  - 系统可能返回任意数量的结果($=N$)
  - 考虑Top k返回的情形(k=0,1,2,…,N)
  - 则每个k的取值对应一个R和P
- →可以计算得到查准率-查全率曲线

查准率-召回率 曲线(Precision - 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%(11点)上对应的查准率，然后描出曲线
- 位于上面的曲线对应的系统结果更好

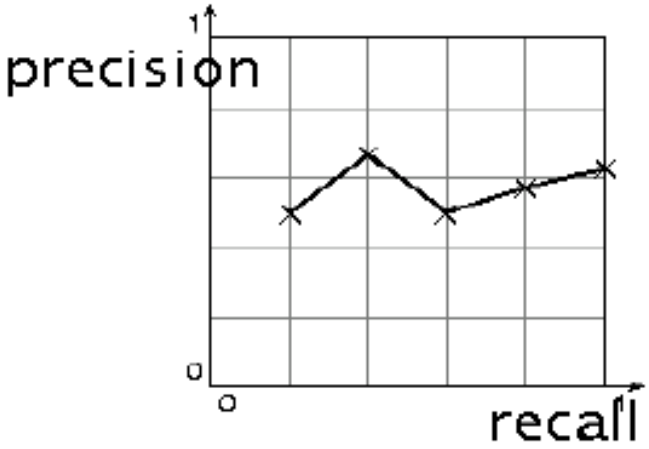
P-R曲线的例子

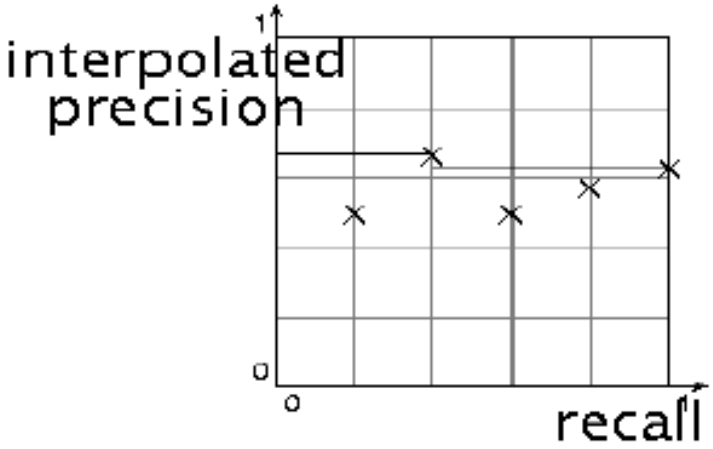
- 某个查询q的标准答案集合为：
  R_q={d_3,d_5,d_9,d_{25},d_{39},d_{44},d_{56},d_{71},d_{89},d_{123}}，$|R_q|$=10
- 某个IR系统对q的检索结果如下：



1. d_{123}√ R=0.1, P=1
2. d_{84} R=0.1, P=0.5
3. d_{56}√ R=0.2, P=0.67
4. d_6 R=0.2, P=0.5
5. d_8 R=0.2, P=0.4
6. d_9√ R=0.3, P=0.5
7. d_{511} R=0.3, P=0.43
8. d_{129} R=0.3, P=0.375
9. d_{187} R=0.3, P=0.33
10. d_{25}√ R=0.4, P=0.4
11. d_{38} R=0.4, P=0.36
12. d_{48} R=0.4, P=0.33
13. d_{250} R=0.4, P=0.31
14. d_{113} R=0.4, P=0.29
15. d_3√ R=0.5, P=0.33

插值查准率





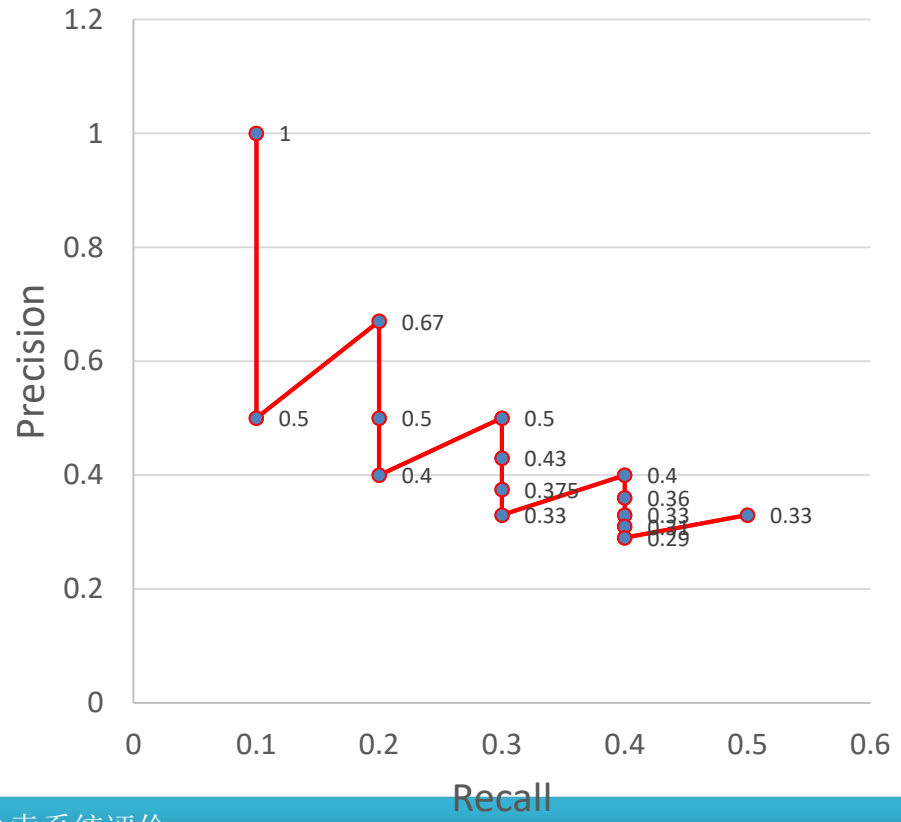
- 原始的曲线常常呈现锯齿状(左图)
  - 如果第(K+1)篇文档**不相关**，则查全率不变，但准确率下降，所以曲线会下降。
  - 如果第(K+1)篇文档**相关**，则查全率和查准率都上升。
- 需要去掉锯齿，进行平滑，采用插值查准率(interpolated precision)，记为P_{interp}
- **在查全率为*r*的位置的插值查准率**：查全率不小于r的位置上的查准率的最大值，即 (见右图)
  - **既每个点往右找最大的P**

$$P_{\text{interp}}(r) = \max_{r' \geq r} P(r')$$

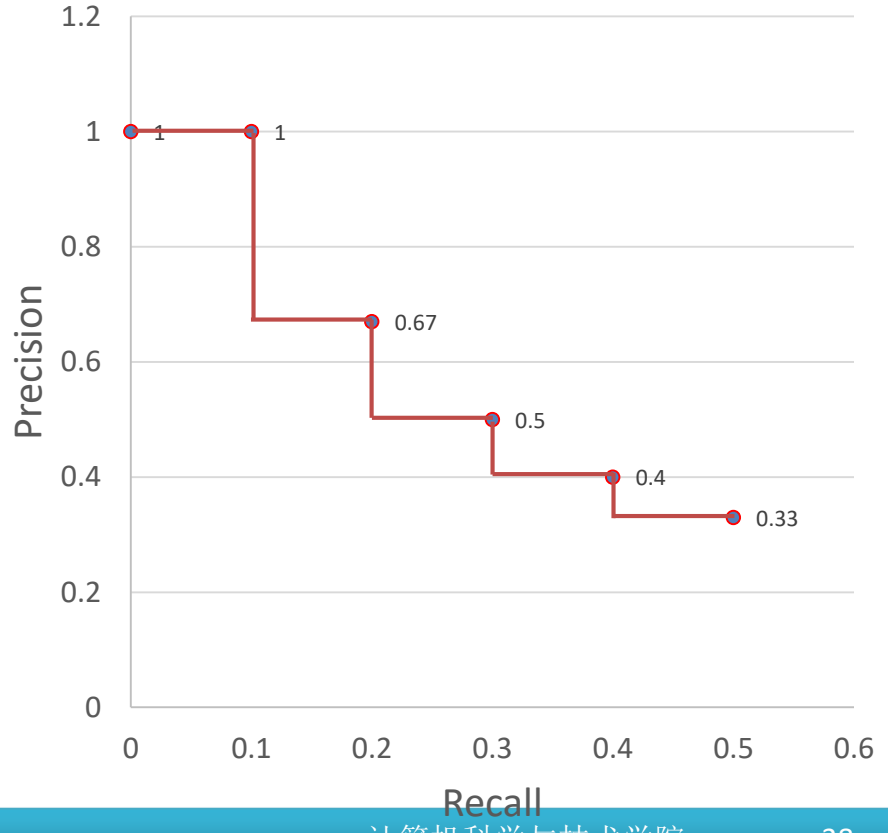
P-R插值例子

1. d_{123} R=0.1, P=1	6. d_9 R=0.3, P=0.5	11. d_{38} R=0.4, P=0.36
2. d_{84} R=0.1, P=0.5	7. d_{511} R=0.3, P=0.43	12. d_{48} R=0.4, P=0.33
3. d_{56} R=0.2, P=0.67	8. d_{129} R=0.3, P=0.375	13. d_{250} R=0.4, P=0.31
4. d_6 R=0.2, P=0.5	9. d_{187} R=0.3, P=0.33	14. d_{113} R=0.4, P=0.29
5. d_8 R=0.2, P=0.4	10. d_{25} R=0.4, P=0.4	15. d_3 R=0.5, P=0.33

P-R



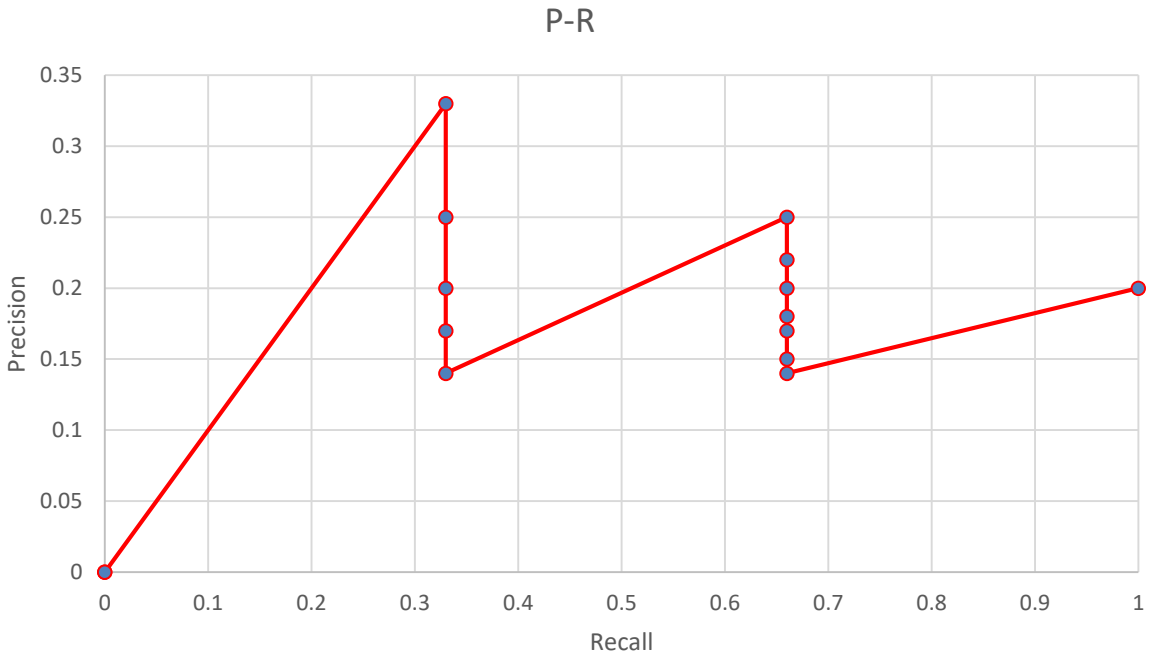
P-R插值

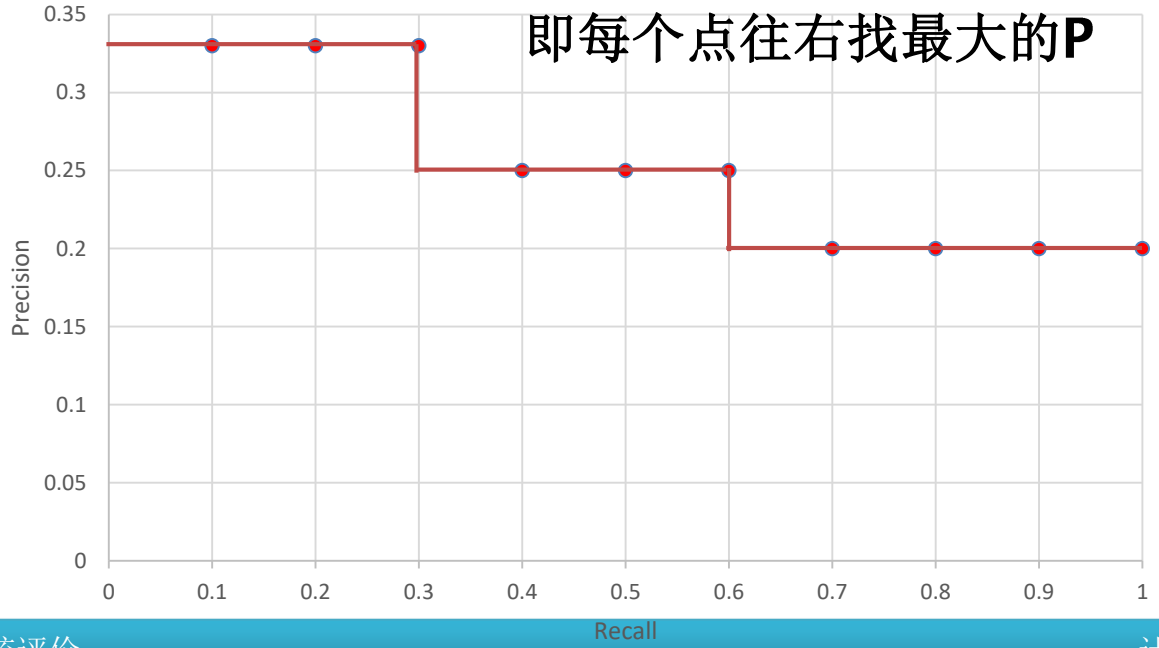


P-R 曲线的插值问题

- 对于前面的例子，假设R_q={d_3,d_{56},d_{129}}
  - 3. d_{56} R=0.33, P=0.33;
  - 8. d_{129} R=0.66, P=0.25;
  - 15. d_3 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

1. d_{123} R=0, P=0
2. d_{84} R=0, P=0
3. d_{56}√ R=0.33, P=0.33
4. d_6 R=0.33, P=0.25
5. d_8 R=0.33, P=0.2
6. d_9 R=0.33, P=0.17
7. d_{511} R=0.33, P=0.14
8. d_{129}√ R=0.66, P=0.25
9. d_{187} R=0.66, P=0.22
10. d_{25} R=0.66, P=0.2
11. d_{38} R=0.66, P=0.18
12. d_{48} R=0.66, P=0.17
13. d_{250} R=0.66, P=0.15
14. d_{113} R=0.66, P=0.14
15. d_3√ R=1, P=0.2





1. d_{123} R=0, P=0
2. d_{84} R=0, P=0
3. d_{56}√ R=0.33, P=0.33
4. d_{6} R=0.33, P=0.25
5. d_{8} R=0.33, P=0.2
6. d_{9} R=0.33, P=0.17
7. d_{511} R=0.33, P=0.14
8. d_{129}√ R=0.66, P=0.25
9. d_{187} R=0.66, P=0.22
10. d_{25} R=0.66, P=0.2
11. d_{38} R=0.66, P=0.18
12. d_{48} R=0.66, P=0.17
13. d_{250} R=0.66, P=0.15
14. d_{113} R=0.66, P=0.14
15. d_{3}√ R=1, P=0.2

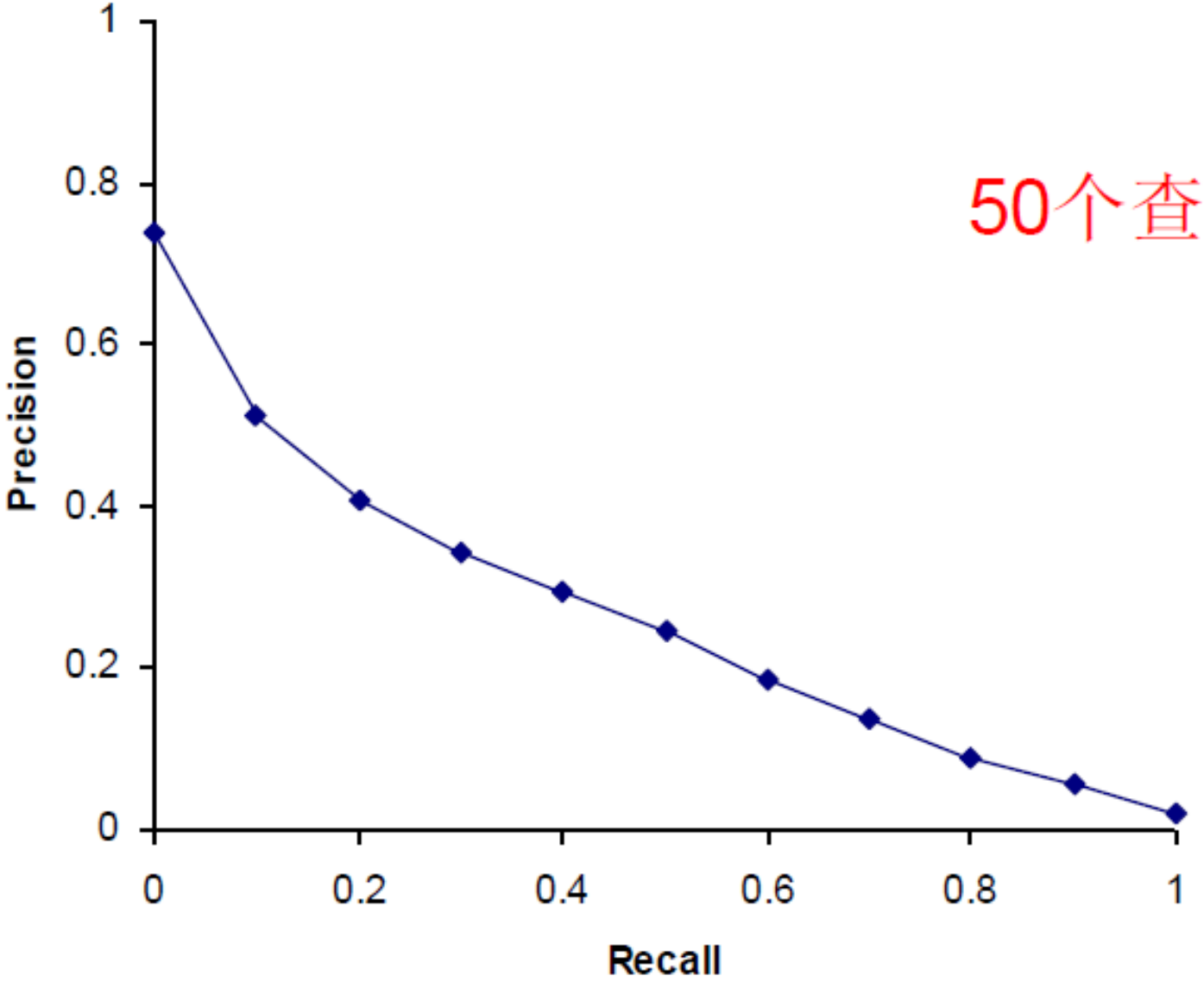
P-R的优缺点

- 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查询的检索结果的优劣

基于P-R曲线的单一指标

- 曲线图虽然好，但是评价标准如果能浓缩成一个数字，就更加清晰明了
  - 固定检索等级的查准率
    - *Precision@k*：前*k*个结果的查准率
    - 对大多数的web搜索是合适的，因为用户看重的是在前几页中有多少好结果
    - 但是这种平均的方式不好，是通常所用指标中最不稳定的
  - 11点平均正确率(11 point average precision)：
    - 对每个信息需求，插值的正确率定义在0、0.1、0.2、…、0.9、1共十一个召回率水平上
    - 对于每个召回率水平，对测试集中多个查询在该点的插值正确率求算术平均。

典型的11点插值正确率-召回率平均曲线



50个查询的平均

更多的评价准则：AP

- 平均查准率(Average Precision, AP)：对不同召回率点上的正确率进行平均
  - **未插值的AP**: 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，等价于6点平均
  - **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

    AP= (1+1+1+1+3/5+3/5+4/10+5/20+5/20+0+0)/11
  - **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，等价于5点平均

不考虑召回率

- Precision@N：在第N个位置上的正确率
  - 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
- bpref、NDCG：后面详细介绍。

系统&查询	1	2	3	4	5
系统1，查询1	d_3 √	d_6 √	d_8	d_{10}	d_{11}
系统1，查询2	d_1 √	d_4	d_7	d_{11}	d_{13} √
系统2，查询1	d_6 √	d_7	d_2	d_9 √	
系统2，查询2	d_1 √	d_2 √	d_4	d_{13} √	d_{14}

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

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

宏平均 vs 微平均

- 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)

课堂练习

- 两个查询q_1、q_2的标准答案数目分别为100个和50个，某系统对q_1检索出80个结果，其中正确数目为40，系统对q_2检索出30个结果，其中正确数目为24，求MacroP/MacroR/MicroP/MicroR：

$P_1 = 40/80 = 0.5,\ R_1 = 40/100 = 0.4$

$P_2 = 24/30 = 0.8,\ R_2 = 24/50 = 0.48$

$MacroP = (P_1+P_2) / 2 = 0.65,$

$MacroR = (R_1+R_2) / 2 = 0.44$

$MicroP = (40+24) / (80+30) = 0.58$

$MicroR = (40+24) / (100+50) = 0.43$

平均查准率均值 Mean Average Precision(MAP)

- 返回每个**相关**文档位置上查准率的平均值，被称为**平均查准率**(AP)
- 对所有查询求宏平均，就得到平均查准率均值(MAP)

$$MAP(Q)=\underbrace{\frac{1}{|Q|}\sum_{j=1}^{|Q|}\underbrace{\frac{1}{m_j}\sum_{k=1}^{m_j}\text{Precision}(R_{jk})}_{\text{AP}}}_{\text{MAP}}$$

  - Q为查询集合， $q_j \in Q$第j个查询
  - q_j所对应的索引里的所有相关文档集合$\{d_1,d_2,\ldots,d_{mj}\}$（标注答案）。对于给定查询$m_j$是常数
  - R_{jk}是查询q_j的返回的列表中的第k个相关文档，Precision(R_{jk})在位置k的查准率（计算时只考虑列表中1...k的文档）
  - 注：k只表示相关文档位置。并不一定是1、2、3、4延续下来，因为相关文档可能在1、4、7位置，只考虑1、4、7

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

已知：查询1相关文档m_1=4，查询2相关文档m_2=3

系统1查询1: P=2/5, R=2/4, F=4/9, AP=(1 +2/2) / 4 = 1/2;

系统1查询2: P=2/5, R=2/3, F=1/2, AP=(1 + 2/5) /3 = 7/15;

系统2查询1: P=2/4, R=2/4, F=1/2, AP=(1 + 2/4) /4 =3/8;

系统2查询2: P=3/5, R=3/3. F=3/4, AP= (1 + 2/2 + 3/4) /3 = 11/12;

系统1：MacroP= (2/5+ 2/5) / 2 = 2/5, MacroR=(2/4 + 2/3) / 2 = 7/12, MacroF=17/36, **MAP**= (1/2 + 7/15) /2 = 29/60,
MicroP=4/10, MicroR=(2+2)/(4+3)=4/7, MicroF= 2*(4/10)*(4/7)/(4/10+4/7) =8/17

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

填空题 4分

设置

P和AP的区别

系统&查询	1	2	3	4	5
系统1，查询1	d3 √	d6 √	d8	d10	d11
系统2，查询1	d3 √	d7	d2	d6 √	d11

已知：查询1相关文档m1=4

系统1：P= [填空1] ,AP= [填空2] 。
系统2：P= [填空3] ,AP= [填空4] 。

正常使用填空题需3.0以上版本雨课堂

作答

P和AP的区别

系统&查询	1	2	3	4	5
系统1，查询1	d3 √	d6 √	d8	d10	d11
系统2，查询1	d3 √	d7	d2	d6 √	d11

已知：查询1相关文档m_1=4

系统1查询1: P=2/5, AP=(1 +2/2) / 4 = 1/2;

系统2查询1: P=2/5, AP= (1 + 2/4) / 4 = 3/8;

从P来看两个系统一样

从AP来看，系统1优于系统2

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**新颖率**(Novelty Ratio)
  - $N=|R_k|/(|R_u|+|R_k|)$，表示系统返回的新相关文档的比例

近几年出现的新的评价指标

- GMAP
- NDCG
- MRR

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%

- 几何平均值

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

- n 查询数目
- 上例 $GMAP_A$ =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

- NDCG，Normalized Discounted Cumulative Gain 归一化折损累计增益
- 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。
- 可以假设，对于返回结果：
  - 相关度级别越高越好
  - 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

CG - Cumulative Gain累计增益

检索结果相关性分数的总和

- 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。
- 只考虑文档相关性级别
- 不考虑文档位置

$$CG = \frac{1}{|Q|}\sum_{j=1}^{|Q|}\sum_{m=1}^{k} R(j,m)$$

j第j个查询

$R(j,m)$是评价人员给出的文档d_m对查询q_j的相关性得分

m是返回结果文档的位置, k是前k个位置

检索结果相关性分数的总和

- 考虑文档相关性级别
- 考虑文档位置: 修改结果越靠前越好，越靠后越不好

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

Discount

商业常用的一个公式，增加相关性级别的权重：

$$DCG = \frac{1}{|Q|}\sum_{j=1}^{|Q|}\sum_{m=1}^{k}\frac{2^{R(j,m)}-1}{\log_2(m+1)}$$

j第j个查询

$R(j,m)$是评价人员给出的文档d_m对查询q_j的相关性级别得分，当$R(j,m)\in\{0,1\}$时，两个公式等价

m是返回结果文档的位置, k是前k个位置

NDCG- Normalized Discounted Cumulative Gain 归一化折损累计增益

同一查询，不同系统返回的结果数量不同，而DCG是一个累加值，没法量化两个不同的系统，因此要进行归一化。

$$NDCG = \frac{DCG}{IDCG}$$

*IDCG*为理想情况下的最大*DCG*值。

$$NDCG(Q,k) = NDCG@k = \frac{1}{|Q|}\sum_{j=1}^{|Q|} Z_j \sum_{m=1}^{k} \frac{R(j,m)}{\log_2(m+1)}$$

或

$$NDCG(Q,k) = NDCG@k = \frac{1}{|Q|}\sum_{j=1}^{|Q|} Z_j \sum_{m=1}^{k} \frac{2^{R(j,m)}-1}{\log_2(m+1)}$$

j第j个查询

$R(j,m)$是评价人员给出的文档d_m对查询q_j的相关性得分

m是返回结果文档的位置, k是前k个位置

Z_j是归一化因子，保证对整个查询NDCG的最大值为1

$$NDCG@k = \frac{1}{|Q|}\sum_{j=1}^{|Q|} Z_j \sum_{m=1}^{k} \frac{R(j,m)}{\log_2(m+1)} \overset{|Q|=1}{\Rightarrow} Z[R(1) + \sum_{m=2}^{k} \frac{R(m)}{\log_2(m+1)}]$$

m	Ground Truth		系统1		系统2	
	Ranking order	*R*(m)	Ranking order	*R*(m)	Ranking order	*R*(m)
1	d_4	2	d_3	2	d_3	2
2	d_3	2	d_4	2	d_2	1
3	d_2	1	d_2	1	d_4	2
4	d_1	0	d_1	0	d_1	0

$$DCG_{GT} = 2 + [\frac{2}{\log_2 3} + \frac{1}{\log_2 4} + \frac{0}{\log_2 5}] = 3.77 \Rightarrow Z = \frac{1}{3.77} \qquad NDCG_{GT} = 1$$

$$DCG_{系统1} = 2 + [\frac{2}{\log_2 3} + \frac{1}{\log_2 4} + \frac{0}{\log_2 5}] = 3.77 \qquad NDCG_{系统1} = Z * 3.77 = 1$$

$$DCG_{系统2} = 2 + [\frac{1}{\log_2 3} + \frac{2}{\log_2 4} + \frac{0}{\log_2 5}] = 3.63 \qquad NDCG_{系统2} = Z*3.63 = 0.96$$

MRR (Mean Reciprocal Rank)

- 倒数排名Reciprocal Rank: 第一个正确答案的位置倒数
- 平均倒数排名MRR:是多个查询结果的平均值

$$\mathrm{MRR} = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{rank_i}$$

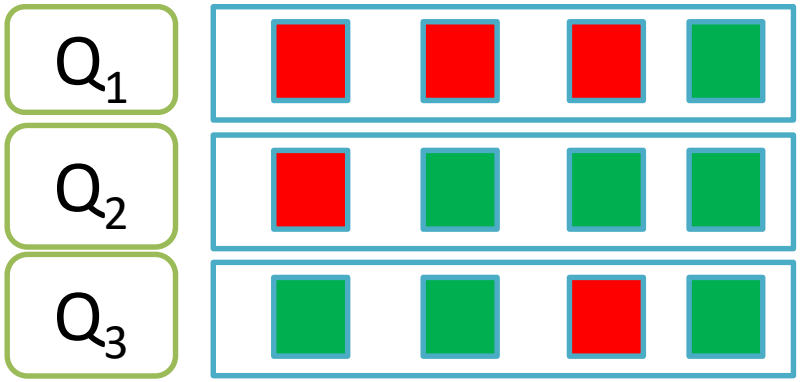
  - $rank_i$第i个查询的第一个正确答案的位次
  - 如果没有正确答案，那么倒数排名=0

填空题 1分

设置

Relevant

None Relevant



MRR= [填空1] （精确到小数点后2位）。

正常使用填空题需3.0以上版本雨课堂

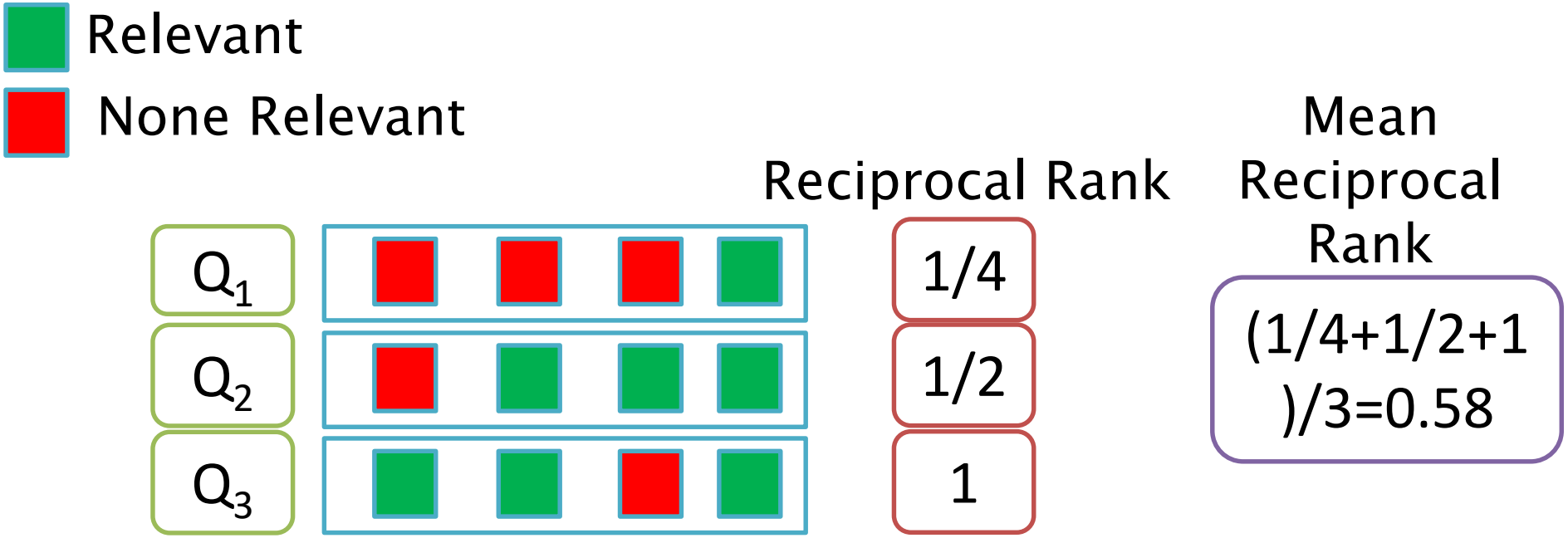
作答

MRR (Mean Reciprocal Rank)

- 倒数排名Reciprocal Rank: 第一个正确答案的位置倒数
- 平均倒数排名MRR:是多个查询结果的平均值

$$\mathrm{MRR} = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{rank_i}$$

  - $rank_i$第i个查询的第一个正确答案的位次
  - 如果没有正确答案，那么倒数排名=0



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？)

目录

- 有关检索评价
- 无序检索结果的评价
- 有序检索结果的评价
- 结果摘要

结果的呈现

- 对与查询相关的检索结果排序后，可以展现一个列表
- 通常情况下，这个列表包含文档的标题和一段摘要(Snippet)

摘要Snippet

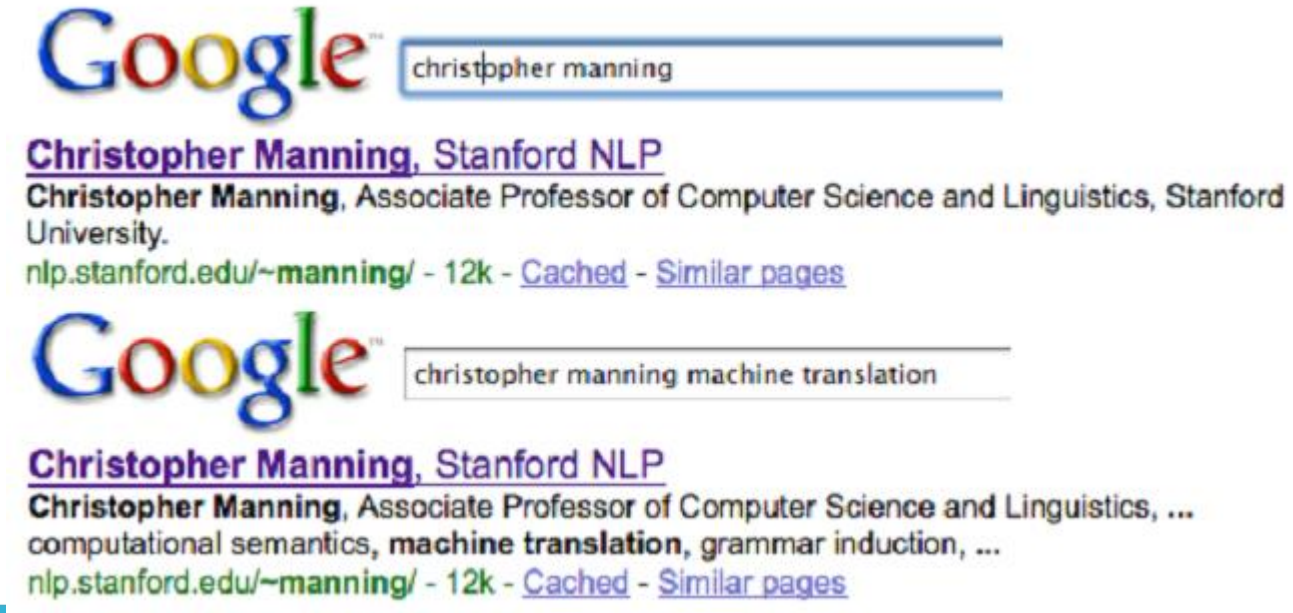
- 标题通常是从文档的元数据中自动抽取出来的
  - 这个描述信息非常重要，用户可以根据它来判断这个文档是不是相关
- 两种基本类型
  - **静态**:不论输入什么查询，文档的静态摘要都是不变的
  - **动态**：而动态摘要依赖于查询，它试图解释当前文档返回的原因

静态摘要Summarization

- 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



动态摘要的生成

- 基于位置索引来构建动态摘要不太合适，至少效率上很低
- 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

本讲小结

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