信息检索与数据挖掘

第6章 检索的评价

课程内容

- 第1章　绪论
- 第2章 布尔检索及倒排索引
- 第3章 词典查找及扩展的倒排索引
- 第4章 索引构建和索引压缩
- 第5章 向量模型及检索系统
- 第6章 检索的评价
- 第7章 相关反馈和查询扩展
- 第8章 概率模型
- 第9章 基于语言建模的检索模型
- 第10章 文本分类
- 第11章 文本聚类
- 第12章 Web搜索
- 第13章 多媒体信息检索
- 第14章 其他应用简介

提纲

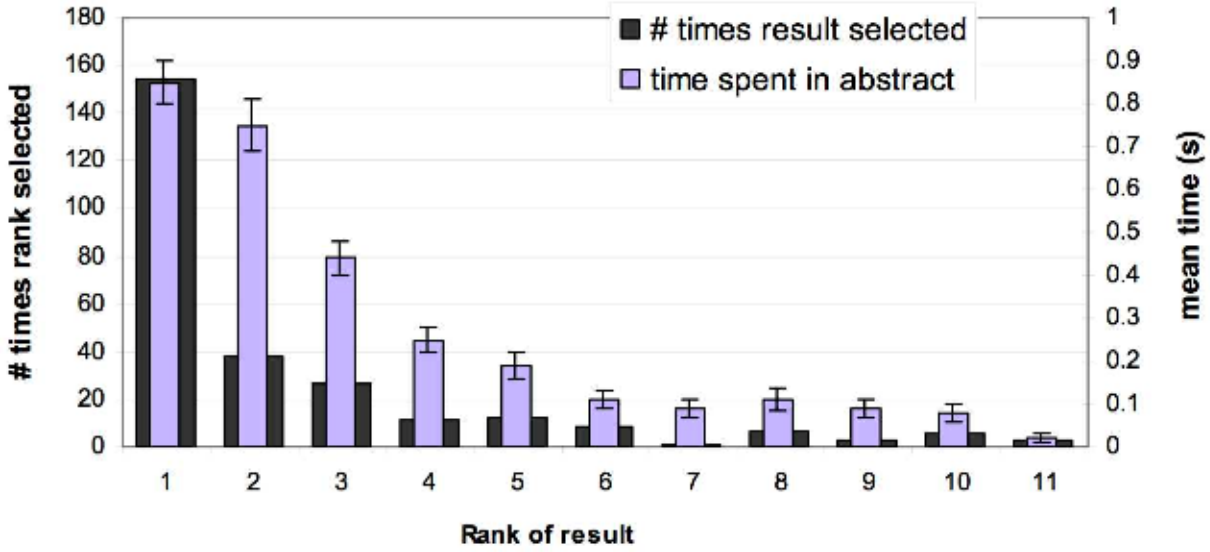
1. 上一讲回顾
2. 检索系统的评价概述
3. 无序检索结果的评价
4. 有序检索结果的评价
5. 为IR系统构建测试集
6. 检索结果的展示

提纲

1. 上一讲回顾
2. 检索系统的评价概述
3. 无序检索结果的评价
4. 有序检索结果的评价
5. 为IR系统构建测试集
6. 检索结果的展示

回顾：检索结果排序的重要性

Looking vs. Clicking



- Users view results one and two more often / thoroughly
- Users click most frequently on result one

Google

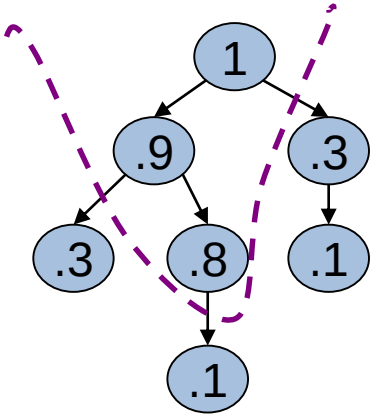
回顾：检索结果排序的实现

精确top K检索加速方法

- 方法一：快速计算余弦

  特例：不考虑查询词项的权重。

- 方法二：堆排序法，N中选K



- 方法三：提前终止计算

  例如将PageRank和余弦相似度线性组合得到文档的最后得分

  net-score(q, d) = g(d) + cos(q, d)

  利用g(d)的有界性，可以提前终止计算

回顾：检索结果排序的实现

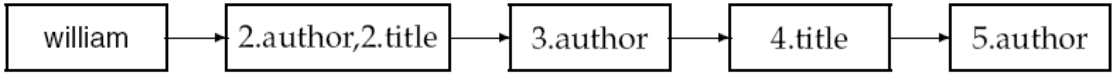
非精确top K检索方法

- 策略一：索引去除(Index elimination)
  - 只考虑那些词项的idf值超过一定阈值的文档
  - 只考虑包含多个查询词项（一个特例是包含全部查询词项）的文档
- 策略二：胜者表
  - 词项t所对应的tf值最高的r篇文档构成t的胜者表
  - 给定查询q，对查询q中所有词项的胜者表求并集，生成集合A。
  - 根据余弦相似度大小从A中选取前top K 个文档
- 策略三：静态得分
  - 衡量文档的权威性
  - 权威性标志举例：Pagerank值、维基百科、报纸上的文章、很多引用的文章、del.icio.us diggs等网站等
- 策略四：影响度排序
  - 将词项t对应的所有文档d按照 $tf_{t,d}$值降序排列

回顾：检索结果排序的实现

非精确top K检索方法

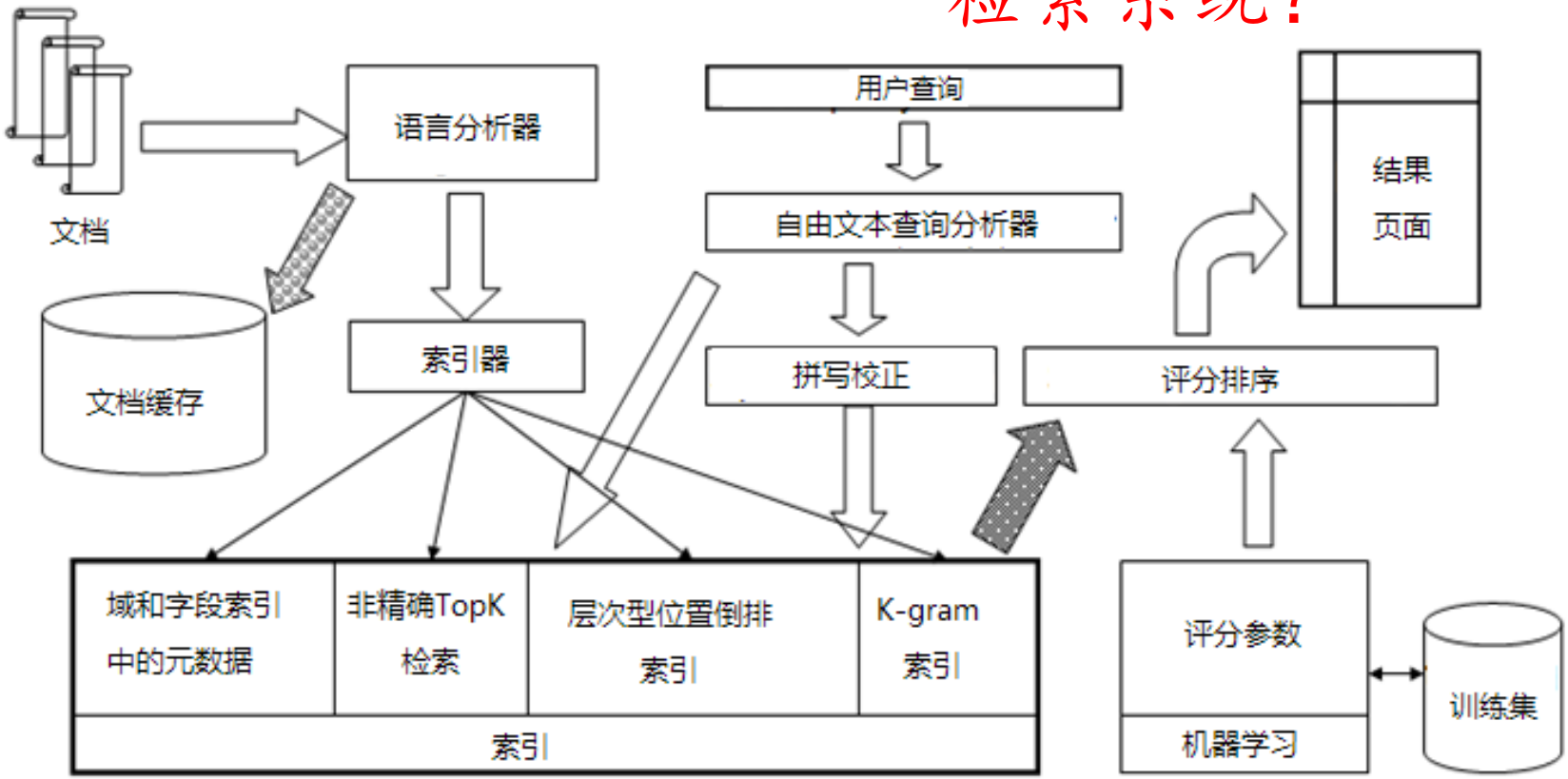
- 策略五：簇剪枝方法
  - 预处理阶段：从N篇文档组成的文档集中随机选出$\sqrt{N}$篇文档（先导者集合）；对于每篇不属于先导者集合的文档，计算与之**距离**最近的先导者。
  - 查询处理：给定查询q，通过与先导者计算余弦相似度，找出和它最近的先导者L；候选集合A包括L及其追随者，然后对A中的所有的文档计算余弦相似度
- 策略六：参数化索引以及域索引



- 策略七：层次索引

回顾：检索系统

如何评价
检索系统？



提纲

1. 上一讲回顾
2. 检索系统的评价概述
3. 无序检索结果的评价
4. 有序检索结果的评价
5. 为IR系统构建测试集
6. 检索结果的展示

关于评价

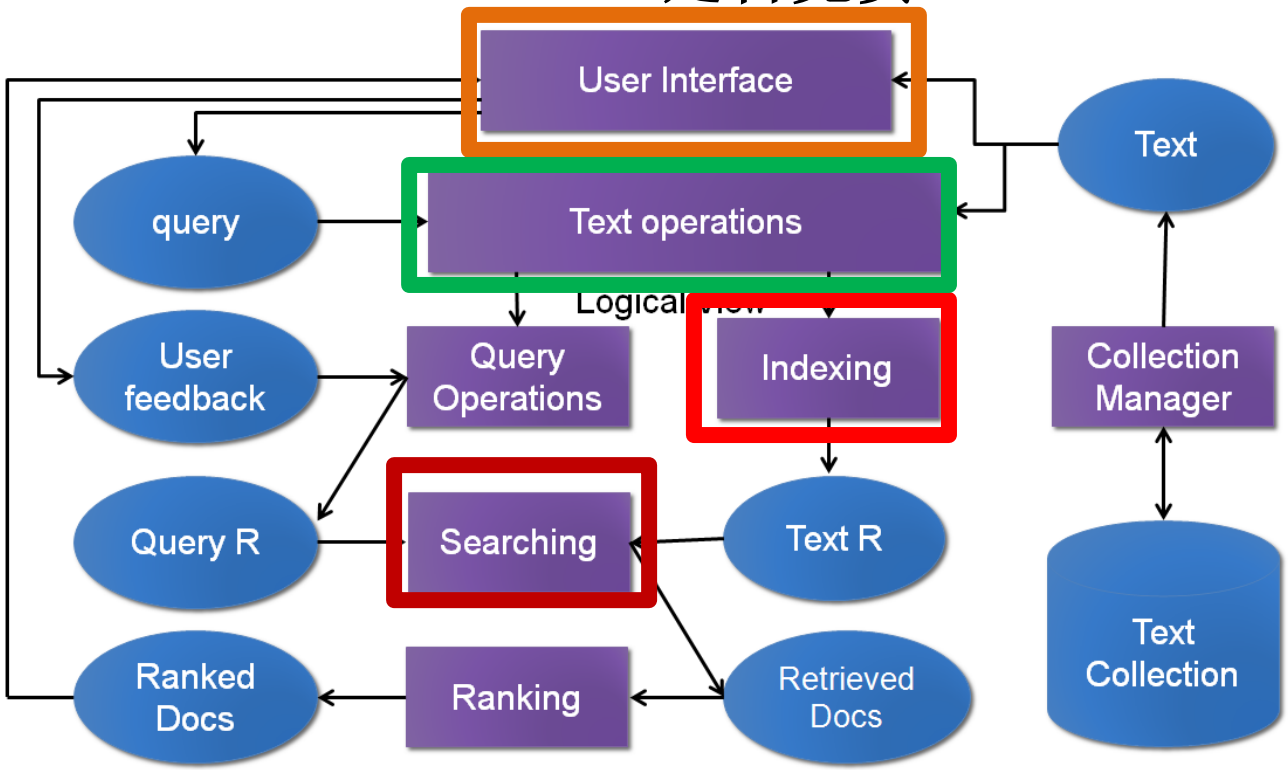
- **评价无处不在，也很必要**
  - 工作、娱乐、招生、找对象
- **评价很难，但是似乎又很容易**
  - 人的因素、标准
- **评价是检验学术进步的唯一标准，也是杜绝学术腐败的有力武器**

为什么要评价IR？

- **通过评价可以判断不同技术的优劣，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。**

搜索引擎的评价

- **建立索引的速度**
  - 每小时索引的文档数量
  - 平均的文档大小
- **搜索的速度**
  - 和索引大小相关
- **查询语言的表达能力**
  - 是否能表达复杂的信息需求
  - 对复杂查询的处理速度
- **流畅和清晰的用户界面**
- **是否免费？**



搜索引擎的评价

- **上述的评价标准都是可以定量的**
  - 我们可以测量速度或者索引大小
- **关键的评价标准：用户满意度**
  - 用户满意度如何定义？
  - 搜索引擎响应速度和索引的覆盖范围是要考虑的因素
  - 但是如果结果不能让用户满意，响应速度再快，也是没有意义的
- **需要一种定量的方法来衡量用户满意度**

如何用客观的 measurement 给出主观的满意度

用户满意度的衡量

- **关键问题：我们要使哪种用户满意？**
  - 根据搜索服务的不同定位而异
- **Web搜索引擎**
  - 用户通过搜索引擎发现自己想要的东西，以后会继续使用这个搜索引擎
    - 可以统计用户的“回头率”
- **电子商务网站**
  - 用户发现自己想要的东西，就会购买
    - 可以统计用户购买所花费时间，以及统计购买的用户占总的搜索的用户的百分比
- **企业：关心“用户的生产力”**
  - 用户使用搜索引擎寻找信息，能节省多少时间？
  - 也需要考虑其他的准则：访问的安全性，访问的广度

满意度是很难衡量的

- **最通常的度量：搜索结果的相关度**
  - 用搜索结果的相关度这个客观度量来替代对满意度的评估

- **→如何衡量相关度？**
- **衡量相关度需要3个要素：**
  1. 评测文档集合
  2. 评测查询集合
  3. 对每个查询的每个返回文档做出“相关”或者“不相关”的评价（有些也可能不是二值的）

信息检索系统的评价

- 需要注意的是，信息需求用查询来表示，但相关性是相对于信息需求而言的，而不是相对于查询而言。

- 例如
  - 信息需求：在降低心脏病发作的风险方面，饮用红葡萄酒是否比饮用白酒更有效？
  - 查询：白酒 红酒 心脏病 有效
  - 在对返回的文档进行评估时，应当考虑是否满足信息需求

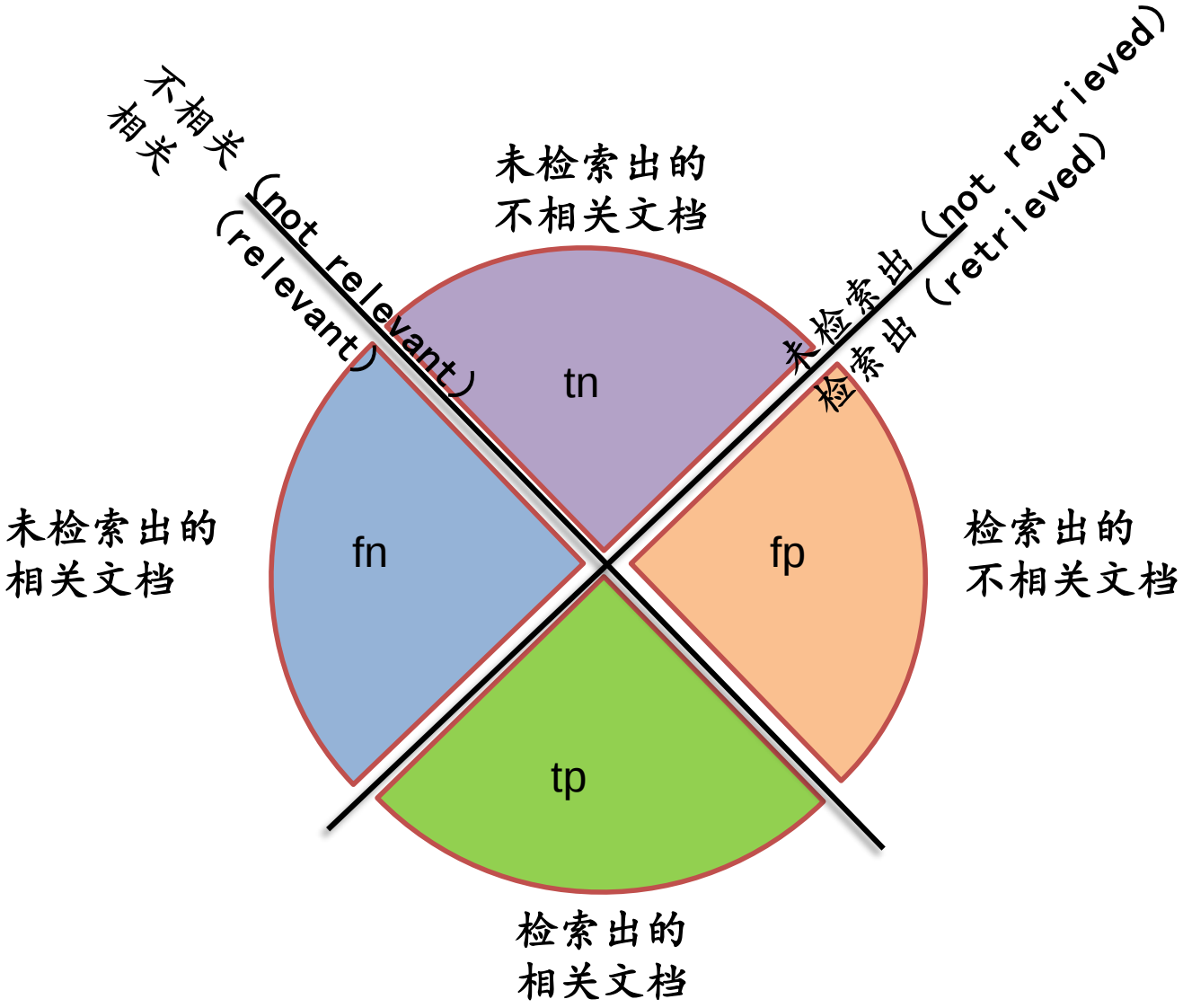
标准的相关度评测准则

- TREC – National Institute of Standards and Technology (美国国家标准技术研究所，NIST)长期维护了一个大规模的IR测试环境
- 评测文档集合包含路透社和其他文档集合
- 在这个框架下定义了很多任务，每个任务都有自己的测试集
- 由人类专家对返回的结果进行“相关”和“不相关”的判定
  - 或者对返回结果的一个子集中的文档进行相关性判定

提纲

1. 上一讲回顾
2. 检索系统的评价概述
3. 无序检索结果的评价
4. 有序检索结果的评价
5. 为IR系统构建测试集
6. 检索结果的展示

按照文档“是否相关”“是否被检索出”划分



与排序无关的检索评价：正确率和召回率

- **正确率/查准率**：返回的相关文档占返回文档总数的百分比
- **召回率/查全率**：返回的相关文档占所有相关文档的百分比。

	Relevant	Nonrelevant
Retrieved	真正例 (true positives, tp)	伪正例 (false positives, fp)
Not Retrieved	伪反例 (false negatives,fn)	真反例 (true negatives,tn)

正确率/查准率Precision $P = tp/(tp + fp)$

召回率/查全率Recall $R = tp/(tp + fn)$

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

Ret = RR+RN

Precision

Recall

Ans = RR+NR

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**
  - Recall=80/100=0.8
  - Precision=80/200=0.4
  - **结论：召回率较高，但是正确率较低**

关于正确率P和召回率R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P和召回率R的讨论(2)

- **虽然Precision和Recall都很重要，但是不同的应用、不用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。

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？**
  - 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

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。

使用正确率/召回率的问题

- 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

- F值是正确率和召回率的加权调和平均数

$$F = \frac{1}{\alpha \frac{1}{P} + (1-\alpha)\frac{1}{R}} = \frac{(\beta^2+1)PR}{\beta^2 P + R}$$

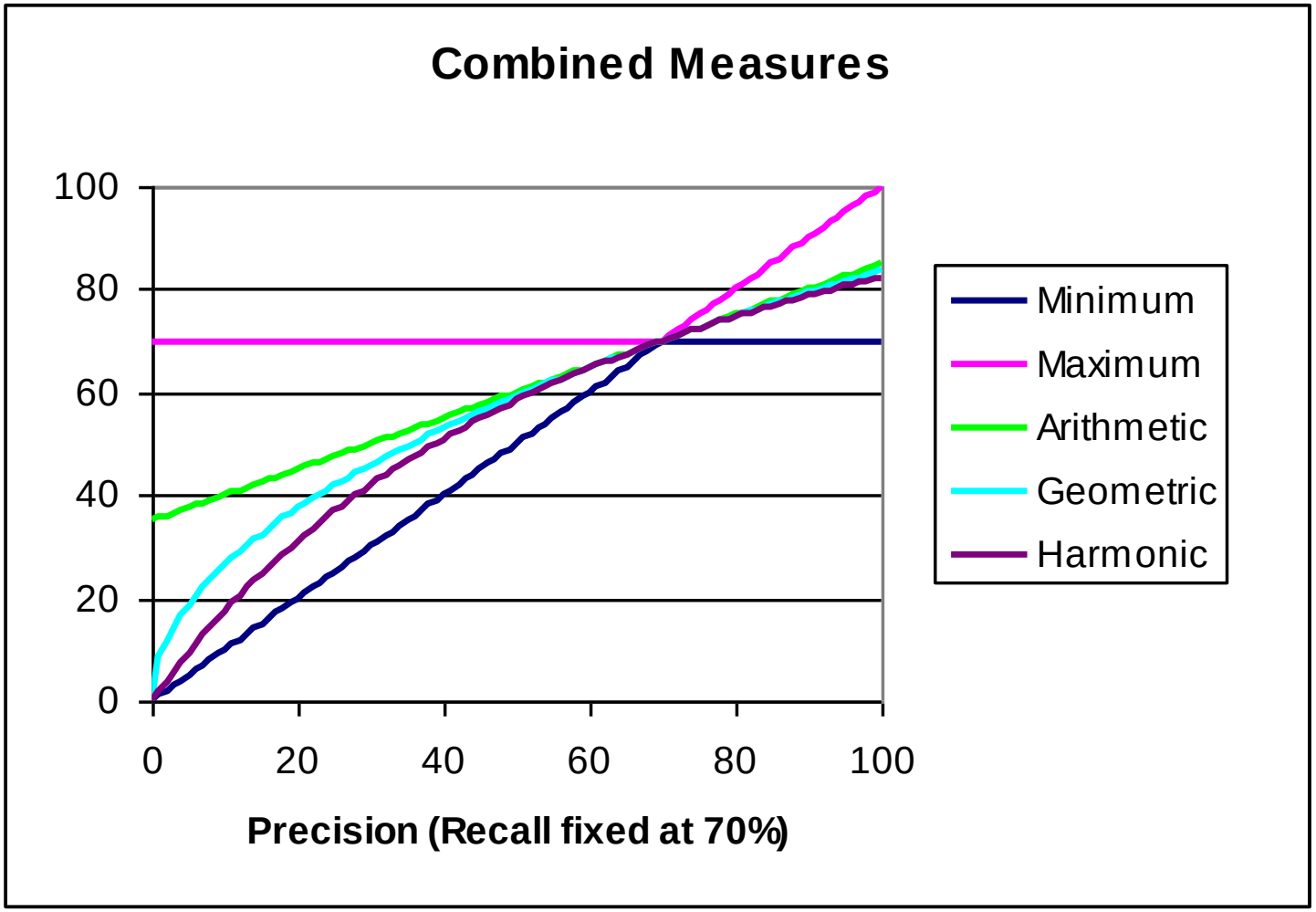
- 通常使用平衡的F_1 值
  - β = 1 or α = ½
- 调和平均比较“保守”
  - 调和平均小于算数平均和几何平均

为什么要使用调和平均？

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
  - 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

$F_{\beta=1}$ 和其他平均数的比较



为什么不使用精确率(Accuracy)？

- 对一个给定的查询，搜索引擎将每篇文档分成“相关”和“不相关”两类。
- 精确率：(tp + tn) / (tp + fp + fn + tn)，被正确分类的文档占总文档的百分比

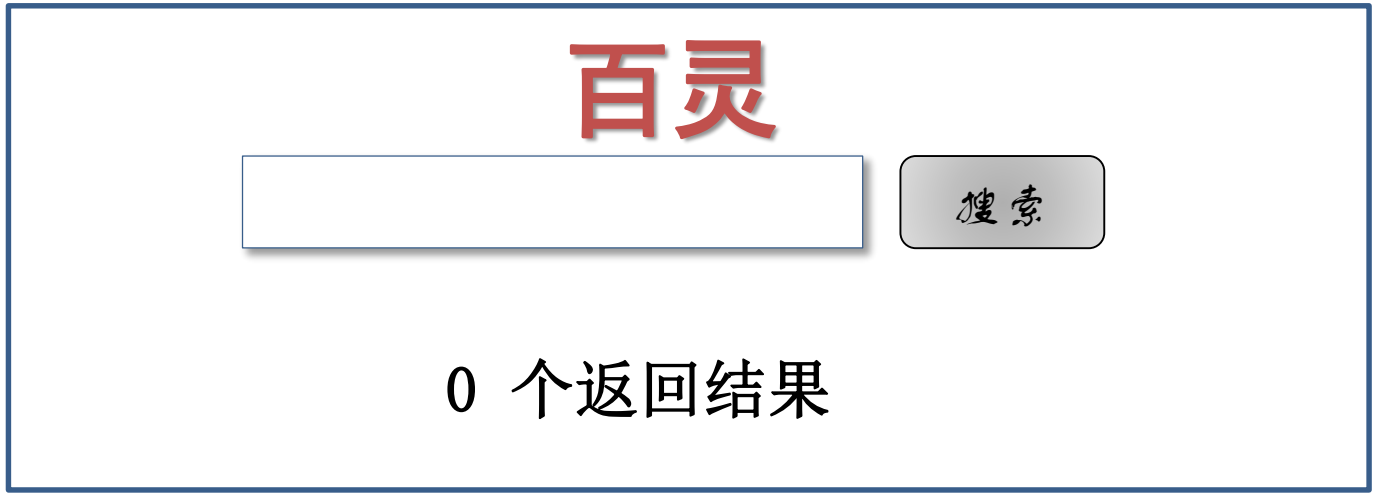
- 精确率是机器学习中模式分类的一个常用评价标准

但是它对信息检索的结果评价不是很有用，

为什么？

精确率不适合IR的原因

- 如何以最低的代价做一个精确率接近100%的搜索引擎？



人们使用搜索引擎，总是希望找到一些有用的信息，即使有些不相关的信息也是可以容忍的

提纲

1. 上一讲回顾
2. 检索系统的评价概述
3. 无序检索结果的评价
4. 有序检索结果的评价
5. 为IR系统构建测试集
6. 检索结果的展示

评价排序后的结果

- P、R、F值都是基于集合的评价方法，它们都利用无序的文档集合进行计算。
  - →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

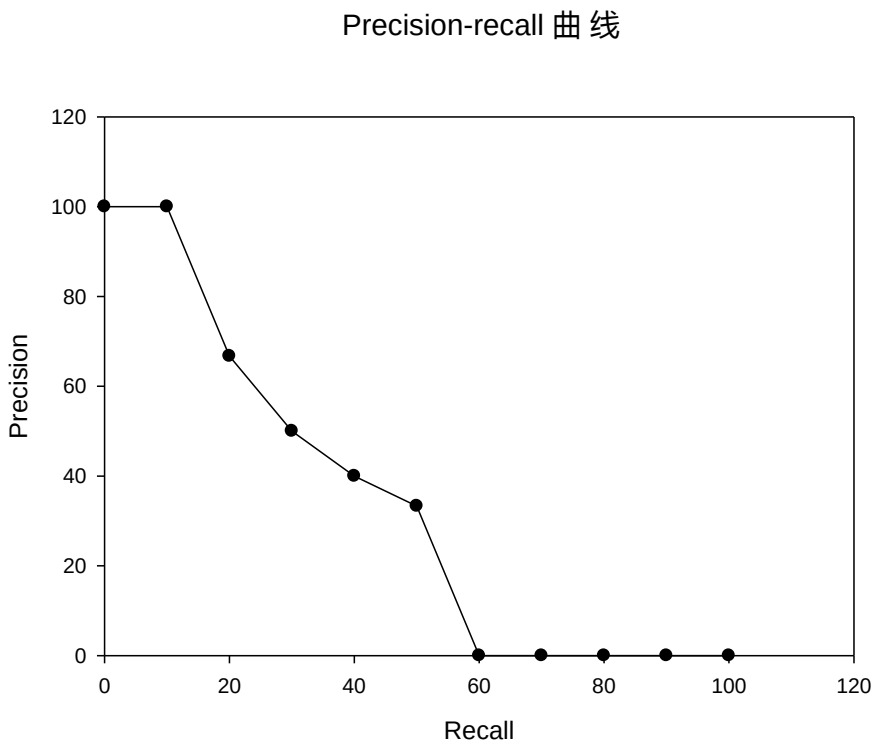
- →可以计算得到正确率-召回率曲线

P-R曲线的例子

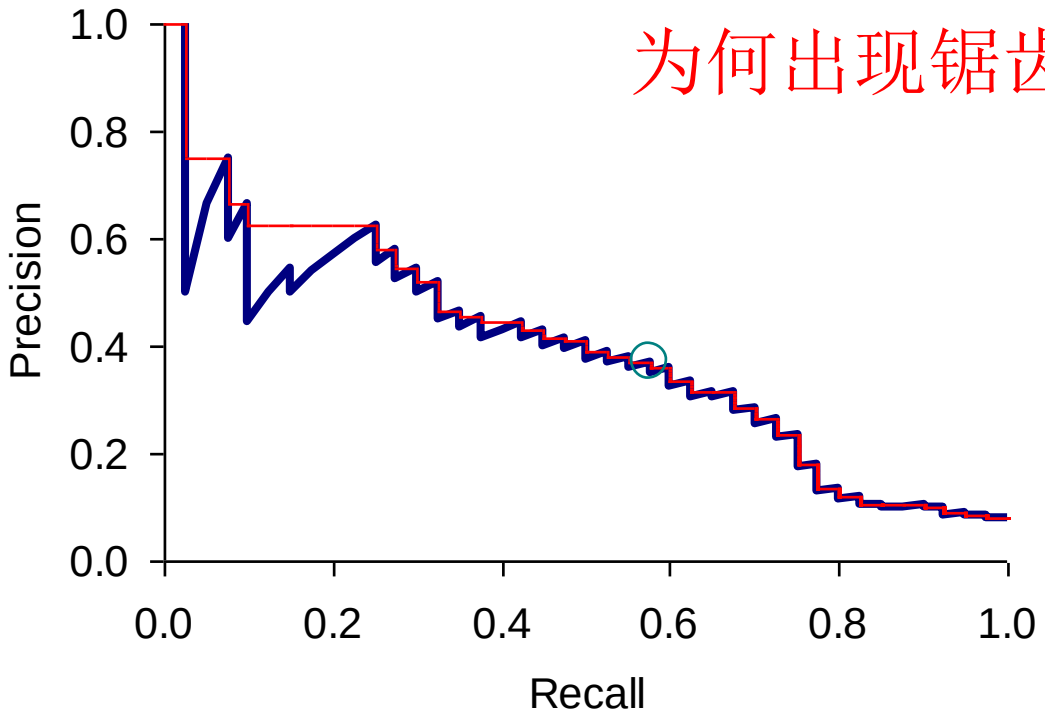
- 某个查询q的标准答案集合为：Rq={d3,d5,d9,d25,d39,d44,d56,d71,d89,d123}
- 某个IR系统对q的检索结果如下：

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

上例的P–R曲线


Precision-recall 曲 线
Precision
Recall
0
20
40
60
80
100
120
0
20
40
60
80
100
120

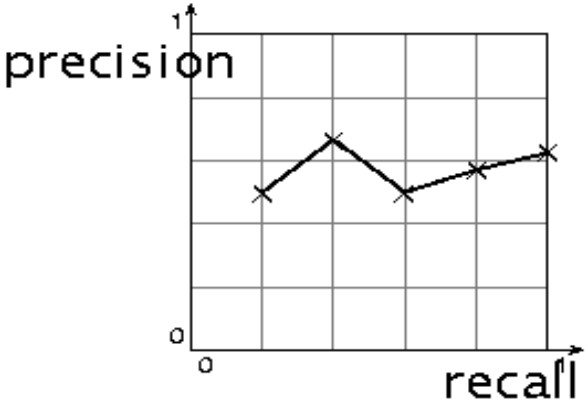
一般的P-R曲线

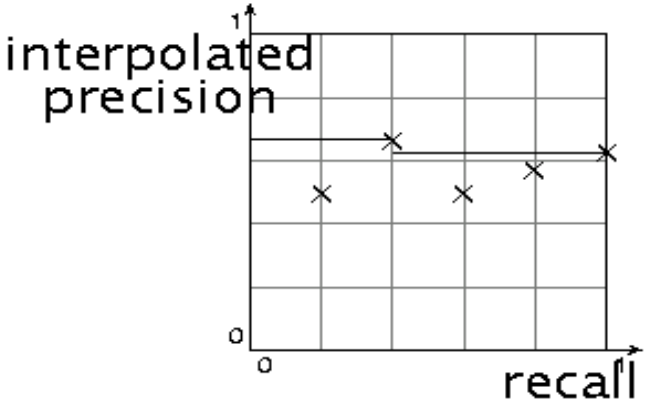


为何出现锯齿的形状？

- 每个点对应top k上的结果（k=1，2，3，4...）
- 插值（红色）：将来所有点上的最高结果
- 插值的原理：如果正确率和召回率都升高，那么用户可能愿意浏览更多的结果，从而提高所看文档中相关文档的比例

插值正确率





原始的曲线常常呈现锯齿状（左图），这是很正常的。因为如果第（K+1）篇文档不相关，则召回率和前k篇文档的召回率是一样的，但是正确率降低了，所以曲线会下降。如果第（K+1）篇文档相关，则召回率和正确率都上升。如此就会出现锯齿状。

我们需要对去掉锯齿，进行平滑。采用**插值正确率**(interpolated precision)，记为p_{interp}

在召回率为r的位置的插值正确率，定义为召回率不小于r的位置上的正确率的最大值，即

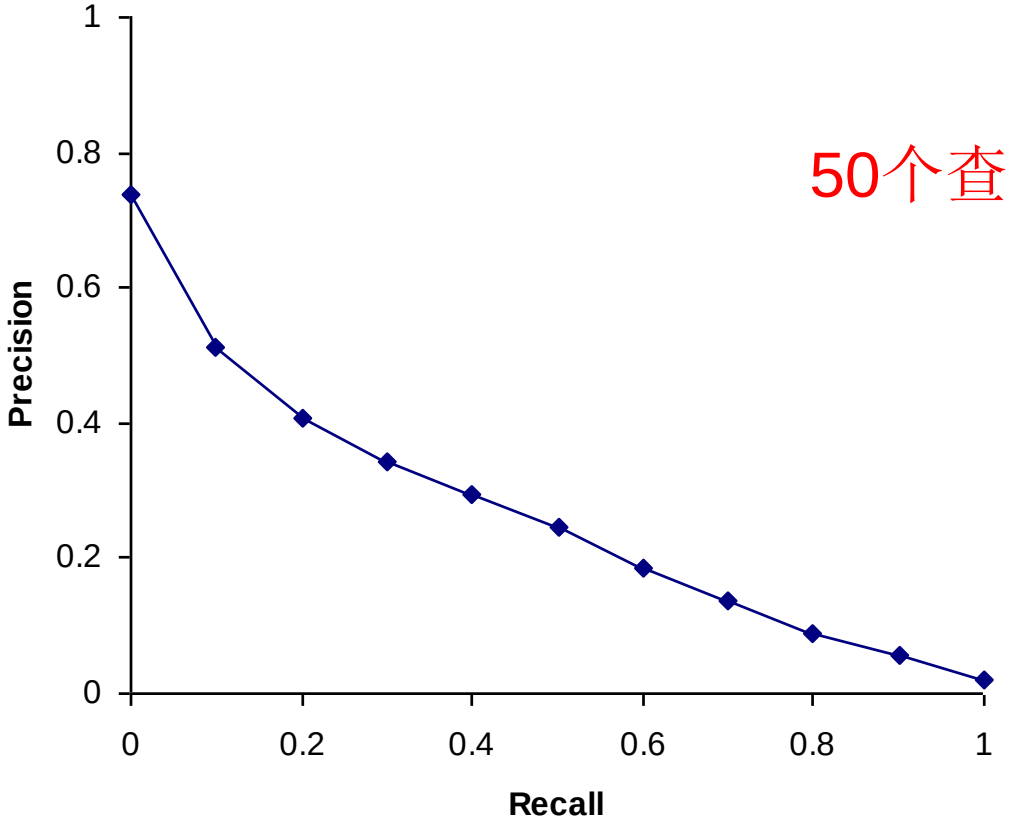
$$p_{interp}(r) = \max_{r' \geq r} p(r')$$

（见右图）

评价

- **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**
  - **固定检索等级的正确率**
    - Precision@k：前k个结果的正确率
    - 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点插值正确率**
    - 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，对测试集中每个信息需求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：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

更多的评价准则：MAP

- **平均正确率均值 Mean Average Precision（MAP）**
  - 在每个相关文档位置上正确率的平均值，被称为平均正确率（AP）
  - 对所有查询求平均，就得到平均正确率均值（MAP）

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

- **参数说明**
  - Q为信息需求， $q_j \in Q$ 所对应的所有相关文档集合为 $\{d_1, d_2, \cdots, d_{mj}\}$，$R_{jk}$是查询$q_j$的返回结果、该结果中包含 $\{d_1, d_2, \cdots, d_k\}$ 而不含有d_{k+1}及以后的相关文档

更多的评价准则：R正确率

- R-Precision
  - 检索结果中，在所有相关文档总数位置上的正确率。如某个查询的相关文档总数为Rel，返回的结果中前|Rel|个中r个是相关文档，则R正确率是r/|Rel|。
  - R正确率能够适应不同的相关文档集的大小
    - 例：Rel=8；r=8。此时R正确率是1，但是P@20=0.4
  - 一个完美的系统的R-precision=1

更多的评价准则：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 $$GMAP = \sqrt[n]{\prod_{i=1}^{n} AP_i} = \exp(\frac{1}{n}\sum_{i=1}^{n} \ln AP_i)$$
- 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

更多的评价准则：NDCG

(Normalized Discounted Cumulative Gain，归一化折损累积增益)

- 针对非二值相关情况下的指标
- 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

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

- R（j,m）是评价人员给出的文档d对查询j的相关性得分，Z_{kj}是归一化因子，保证对完美系统NDCG的值为1，m是返回文档的位置

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？)

提纲

1. 上一讲回顾
2. 检索系统的评价概述
3. 无序检索结果的评价
4. 有序检索结果的评价
5. 为IR系统构建测试集
6. 检索结果的展示

常用的测试集

Collection	*NDocs*	*NQrys*	*Size (MB)*	*Term/Doc*	*Q-D RelAss*
ADI	82	35			
AIT	2109	14	2	400	>10,000
CACM	3204	64	2	24.5	
CISI	1460	112	2	46.5	
Cranfield	1400	225	2	53.1	
LISA	5872	35	3		
Medline	1033	30	1		
NPL	11,429	93	3		
OSHMED	34,8566	106	400	250	16,140
Reuters	21,578	672	28	131	
TREC	740,000	200	2000	89-3543	» 100,000

从文档集合如何构建测试集

- 需要“用于测试的查询”和“相关性的判定”
  - 用于测试的查询
    - 必须和测试文档集合有密切关系
    - 最好由领域的专家设计
    - 随机的查询并不好
  - 相关性的判定
    - 人工判定耗时较长
    - 使用一组人进行判定是否是最好的方式?

用户判定的有效性

- 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

相关性判定之间的一致性

- Kappa统计量
  - 衡量不同人做出的相关性判定之间的一致性
  - 对随机一致性比率的简单校正
- Kappa = [P(A) - P(E)] / [1 - P(E)]
- P(A) - 实际观察到的一致性判断比率
- P(E) - 随机情况下所期望的一致性判断的比率
- Kappa = 0 和随机判断的情况一样，Kappa = 1 不同人做出的相关性判定完全一致.
- *k在* [2/3, 1.0]时，判定结果是可以接受的
- 如果k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

		第二个人的相关性判定		
		Yes	No	Total
第一个人的相关性判定	Yes	300	20	320
	No	10	70	80
	Total	310	90	400

观察到的两个人的一致性判断比率

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

边缘统计量

$P(nonrelevant) = (80 + 90)/(400 + 400) = 170/800 = 0.2125$

$P(relevant) = (320 + 310)/(400 + 400) = 630/800 = 0.7878$

两个人的随机一致性比率

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

Kappa统计量

$\kappa = (P(A) - P(E))/(1 - P(E)) = (0.925 - 0.665)/(1 - 0.665) = 0.776$ (still in acceptable range)

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 . . .
- . . . 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) . . .
  - 当然，如果考虑点击历史的整体情况会相当可靠
- 举例: A/B 测试

A/B 测试

- 目标: 测试某个新引入的独立的创新点
- 先决条件：大型的搜索引擎已经在上线运行
- 方法：
  - 很多用户使用老系统，将一小部分(如 1%)流量被随机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，比如判断第一个结果的点击率是否有提升
  - 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

提纲

1. 上一讲回顾
2. 检索系统的评价概述
3. 无序检索结果的评价
4. 有序检索结果的评价
5. 为IR系统构建测试集
6. 检索结果的展示

结果摘要

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

John McCain
John McCain 2008 - The Official Website of **John McCain**'s 2008 Campaign for President ... African American Coalition; Americans of Faith; American Indians for **McCain**; Americans with ...
www.**johnmccain**.com · Cached page

JohnMcCain.com - **McCain**-Palin 2008
John McCain 2008 - The Official Website of **John McCain**'s 2008 Campaign for President ... African American Coalition; Americans of Faith; American Indians for **McCain**; Americans with ...
www.**johnmccain**.com/Informing/Issues · Cached page

John McCain News- msnbc.com
Complete political coverage of **John McCain**. ... Republican leaders said Saturday that they were worried that Sen. **John McCain** was heading for defeat unless he brought stability to ...
www.msnbc.msn.com/id/16438320 · Cached page

John McCain | Facebook
Welcome to the official Facebook Page of **John McCain**. Get exclusive content and interact with **John McCain** right from Facebook. Join Facebook to create your own Page or to start ...
www.facebook.com/**johnmccain** · Cached page

摘要

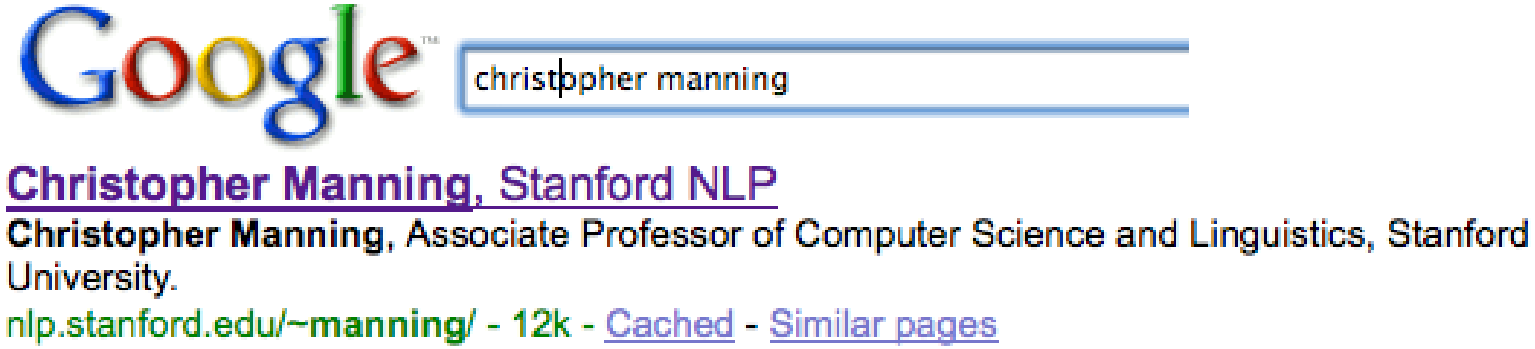
- **标题通常是从文档的元数据中自动抽取出来的**
  - 这个描述信息非常重要，用户可以根据它来判断这个文档是不是相关
- **两种基本的摘要方法**
  - 静态：一个文档的摘要是固定的，与查询无关
  - 动态：与查询相关。摘要说明了为什么这篇文档和查询相关

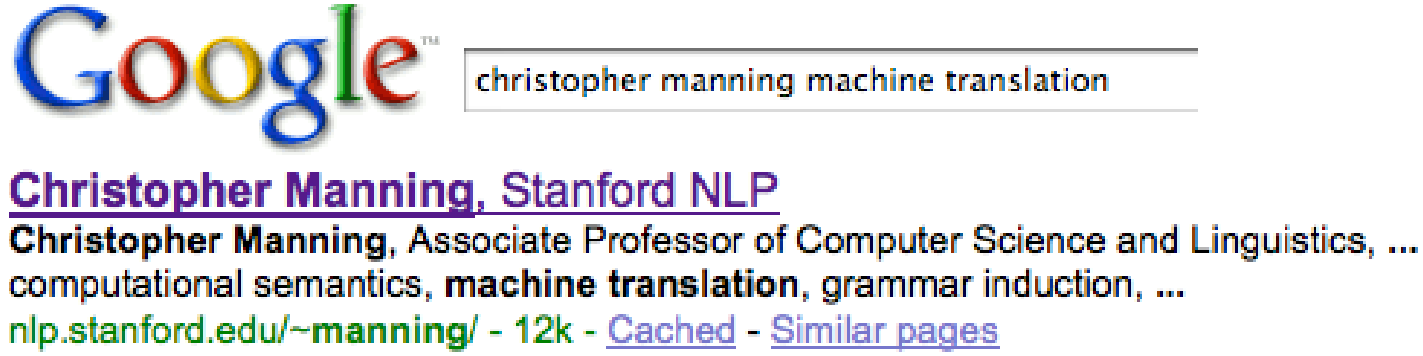
静态摘要

- 在典型的系统中，静态摘要是文档的一个子集
- 最简单的方法：文档的前若干个词汇
  - 在建立索引的时候就缓存好
- 更复杂的方法：从文档中抽取一些关键的句子
  - 用简单的自然语言处理的方法对句子进行打分
    用打分最高的几个句子组成摘要
- 最复杂的方法：用自然语言处理的方法合成摘要
  - 在IR系统中几乎不用

动态摘要

- 显示文档中包含查询词的一句或者几句文字
  - “KWIC”片段： Keyword in Context presentation





动态摘要相关技术

- **快速在文档中寻找包含查询词的“窗口”（范围）**
- **根据查询对文档中上述窗口打分**
  - 用多种特征，如窗口的长度，在文档中的位置，等等
  - 用一个打分函数融合多种特征
- **评价的挑战：对摘要的评价**
  - 相关度的两两比较比单个文档的相关度判定简单

快捷链接

- 对导航性的查询，例如搜索“中国科学技术大学”，将会在页面上显示一些导航链接

中国科学技术大学

找到约 1,410,000 条结果（用时 0.17 秒）

中国科学技术大学

中国科学院所属的一所以前沿科学和高新技术为主、兼有以科技为背景的管理和人文学科的综合性全国重点大学。

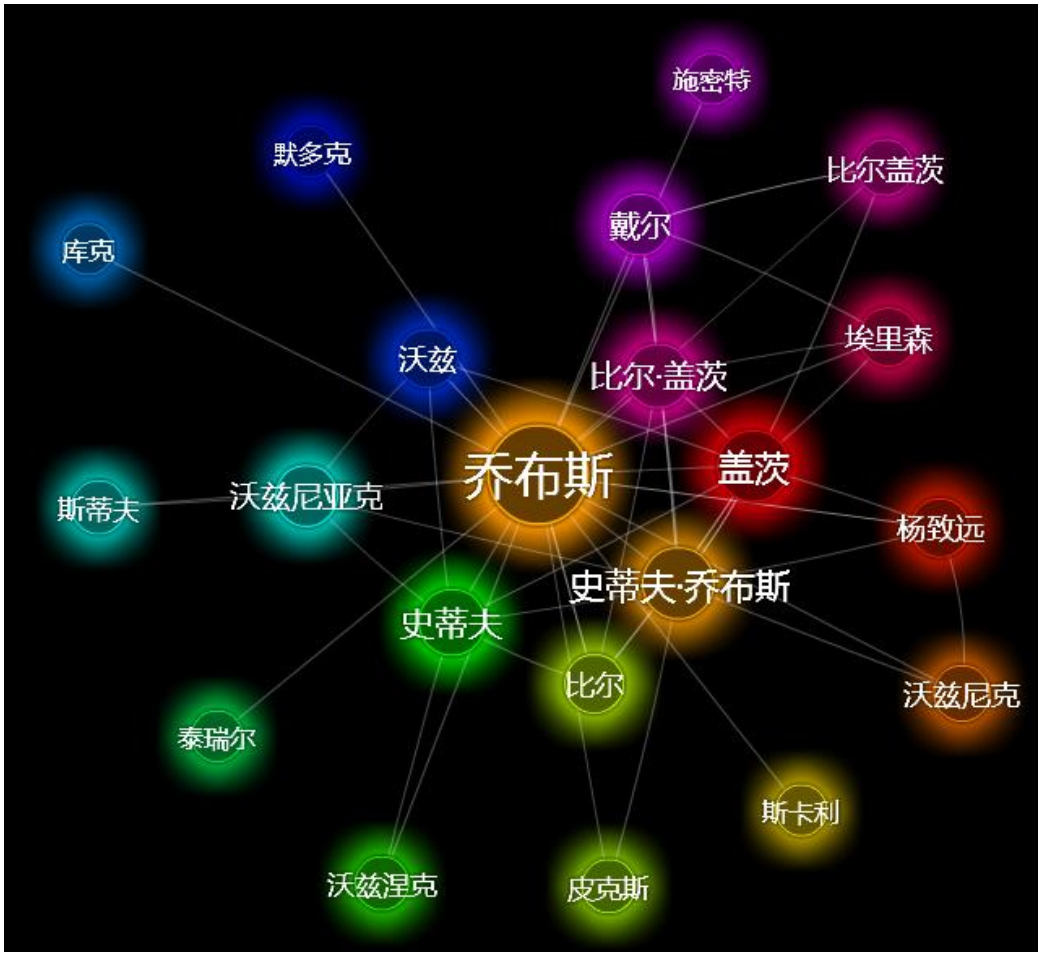
www.ustc.edu.cn/ - 网页快照 - 类似结果

电子邮件	公共服务
研究生教育	生命科学学院
招生在线	学校简介
本科生教育	热点连接

ustc.edu.cn站内的其它相关信息 »

其他的展现方式

- 人立方 http://renlifang.msra.cn/GuanxiMap.aspx



课后练习

- 习题8-8
- 习题8-9

谢谢大家！