

信息检索方法研究



Krse Lee
21551081

信息检索



- ❖ 信息检索是从大规模非结构化数据(通常是文本)的集合中找出满足用户信息需求的资料的过程。
- ❖ Web搜索引擎：Google，百度（网络爬虫+检索系统）
- ❖ 个人电脑搜索
- ❖ 面向企业、机构、特定领域的搜索：站内搜索

布尔检索模型



and or not

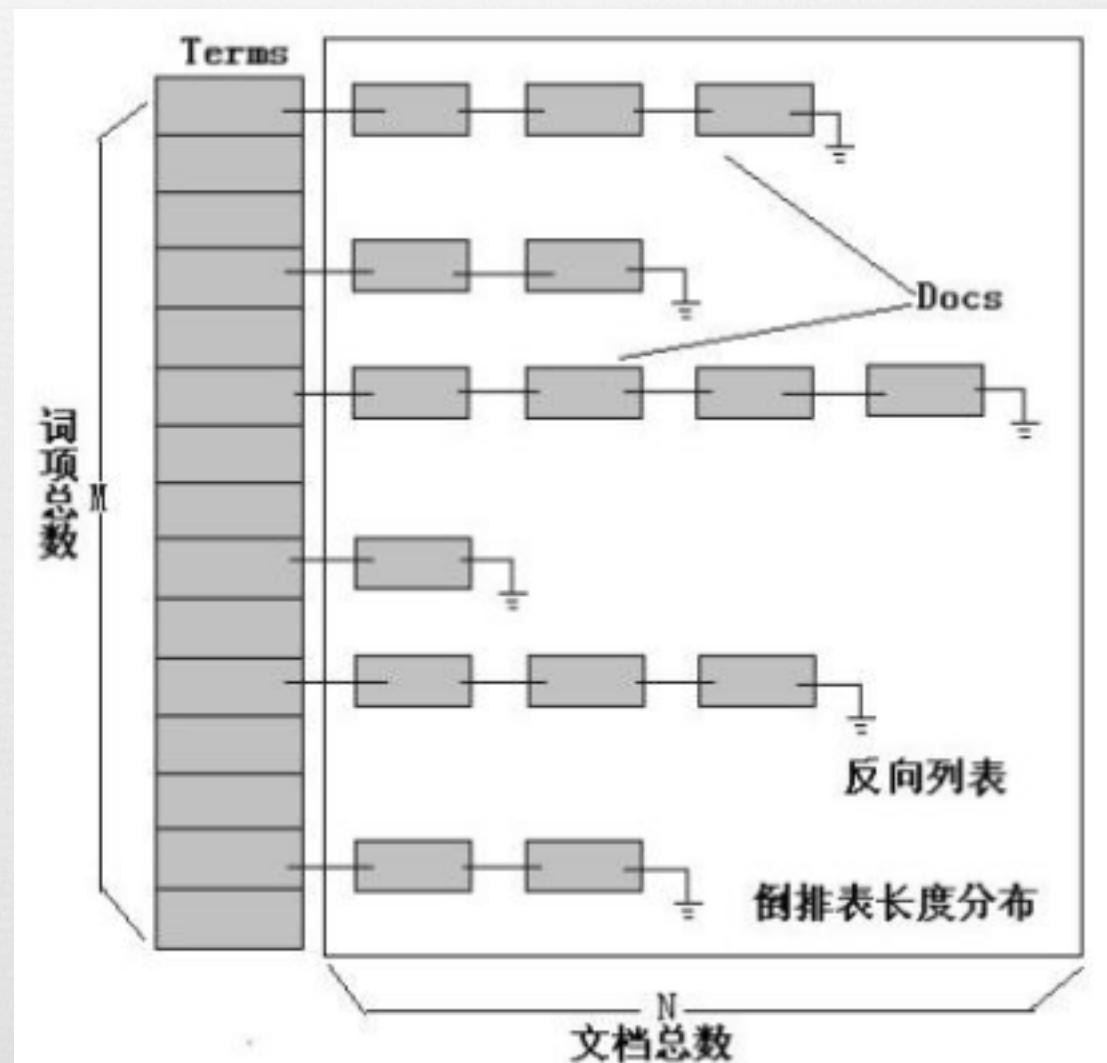
布尔检索



- ❖ 通过 AND、OR 及 NOT 等逻辑操作符将词项连接起来的查询。
- ❖ 例如：“信息” and “检索”，“机器” and “学习”
- ❖ 过程：词条化、建立倒排索引、处理查询。

倒排索引

- 词项词典 + 所有词项的倒排记录
- 倒排记录中记录的是该词项出现的位置



查询处理



- ❖ 处理“信息” and “检索”
- ❖ 合并算法，取交集，找到1, 9, 86三个文档

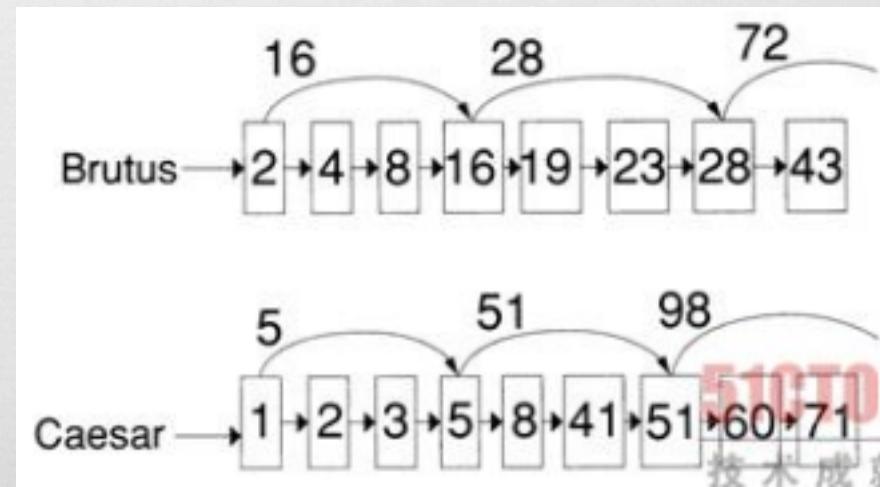
```
Intersect (p1, p2)
answer
while p1 != NIL and p2 !=NIL
do if docID(p1) = docID(p2)
    then Add( answer, docID(p1) )
        p1 .next(p1)
        p2 .next(p2)
    else if docID(p1) < docID(p2)
        then p1 .next(p1)
    else p2 .next(p2)
return answer
```

信息	1	6	9	50	86
检索	1	9	86		
机器	3	6	34	72	
学习	1	6	72		
自然	3	50			
语言	3	29	50	67	
处理	3	43	50	70	86

跳表



- ❖ 快速的跳过部分列表
- ❖ 当查找 Brutus 和 Caesar 时,查找到 8 这个记录后,Caesar 到了 41,Brutus 需要连续比较 4 次才会来到 43 这个记录,如果在 Brutus 的 14 记录上增加一个指针指向 28,那么可以跳过 19、23 的比较,从而加快查询效率。



词项集合的确定



Friends, Romans, countrymen. So let it be with Caesar...

❖ 词条化：拼写问题 Friends Romans countrymen So ...

❖ 去除停用词：停用词表 Friends Romans countrymen

❖ 词项归一化：显示等价类和隐式等价类 Friends roman countrymen

❖ 词干还原和词形归并：去除前后缀

向量空间模型



文档的向量化



- ❖ $d = \{t_1, t_2, t_3, \dots, t_4\}$
- ❖ 将一篇文档看成是一个向量,其中每个分量都对应词典中的一个词项,采用权重计算公式计算出词项的权重值作为该分量的大小,
- ❖ **词袋模型:** 词与词之间的顺序和关联关系,只保留出现的次数。
- ❖ 高频词权重大, 低频词权重小。

Tf-IDf权重



- ❖ and, if, so, but, I, is, do这类词几乎在所有的文档中都会出现，并且频率很高，降低查询的用户体验
- ❖ 逆文档频率**idf**:
 - ❖ df: 文档频率; N: 文档数目
 - ❖ 一个词项在越多文档中出现, idf就越低
- ❖ **Tf-IDf**
 - ❖ Tf: 词频
 - ❖ 削弱常用高频词所占权重
$$tf - idf_{t,d} = tf_{t,d} \times idf_t$$

余弦相似度



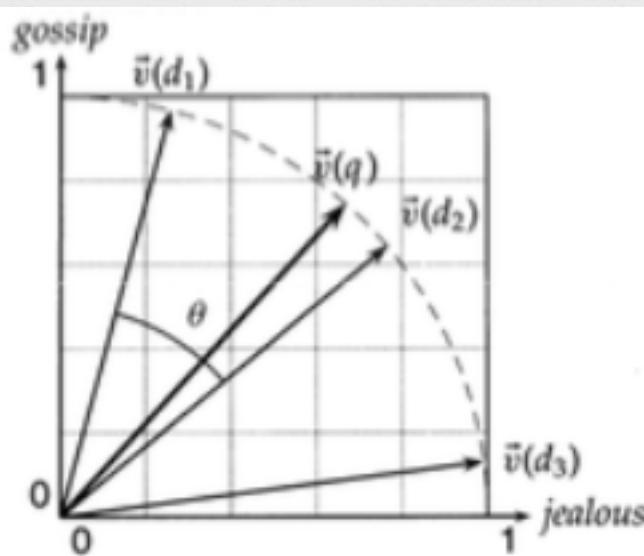
≈ 向量内积：

$$\{a_1, b_1, c_1\}^* \{a_2, b_2, c_2\} = \{a_1 * a_2, b_1 * b_2, c_1 * c_2\}$$

≈ 余弦相似度

$$sim(d_1, d_2) = \frac{\vec{V}(d_1) \cdot \vec{V}(d_2)}{|\vec{V}(d_1)| \times |\vec{V}(d_2)|}$$

≈ 单位向量：平衡长文档和短文档之间的权重失衡



查询处理



对于查询q

$$score(q, d) = \frac{\vec{V}(q) \cdot \vec{V}(d)}{|\vec{V}(q)| \times |\vec{V}(d)|}$$

信息	N/df _{信息}	Tf _{信息,d1}	Tf _{信息,d2}	Tf _{信息,d5}	Tf _{信息,d9}
检索	N/df _{检索}	Tf _{检索,d1}	Tf _{检索,d3}	Tf _{检索,d5}	Tf _{检索,d7}
机器	N/df _{机器}	Tf _{机器,d3}	Tf _{机器,d8}		
学习	N/df _{学习}	Tf _{学习,d4}	Tf _{学习,d8}	Tf _{学习,d9}	
自然	N/df _{自然}	Tf _{自然,d1}	Tf _{自然,d3}		
语言	N/df _{语言}	Tf _{语言,d3}			
处理	N/df _{处理}	Tf _{处理,d3}	Tf _{处理,d7}	Tf _{处理,d9}	

CosineScore(q)

float Scores[N] = 0

Initialize Length[N]

for 每个查询词项 t

do 从倒排索引中取出 t 的信息，计算 t 在查询向量中的权重 W_{t,q}
for 倒排索引中的每个文档、tf 值对(d, tf_{t,d})

do Scores[d] += W_{f_{t,d}} * W_{t,q}

读取 Length[d]

for 每一个 d

do Scores[d] = Scores[d] / Length[d]

return Scores[]中前 K 得分的值

快速评分和排序



- ❖ 相对权重替代绝对权重，忽略查询词项权重
- ❖ 乘法变加法，浮点形变整形

CosineScore(q)

```
float Scores[N] = 0
Initialize Length[N]
for 每个查询词项 t
    do 从倒排索引中取出 t 的信息，计算 t 在查询向量中的权重  $W_{t,q}$ 
    for 倒排索引中的每个文档、tf 值对( d, tft,d )
        do Scores[d] +=  $W_{t,q}$ 
读取 Length[d]
for 每一个 d
    do Scores[d] = Scores[d] / Length[d]
return Scores[] 中前 K 得分的值
```

The screenshot shows a Google search results page for the query "信息检索". The top result is a link to Baidu's百科 entry on Information Retrieval. Below it, there are several other links related to information retrieval, including entries from Wikipedia and various MOOC platforms like iCourse163.

Rank	Title	Description
1	信息检索（一种信息技术）_百度百科	找到约 2,370,000 条结果 (用时 0.29 秒)
2	信息检索- 维基百科, 自由的百科全书	https://zh.wikipedia.org/zh-cn/信息检索 转为简体网页
3	信息检索-中国大学MOOC(慕课)	www.icourse163.org/course/wzu-29001 转为简体网页
4	信息检索-MBA智库百科	wiki.mbalib.com/wiki/信息检索 转为简体网页
5	哈尔滨工业大学社会计算与信息检索研究中心-理解语言, 认...	ir.hit.edu.cn/ 转为简体网页

非精确返回前K篇文档



❖ 索引去除

❖ 只考虑 idf 值超过一定阀值的词

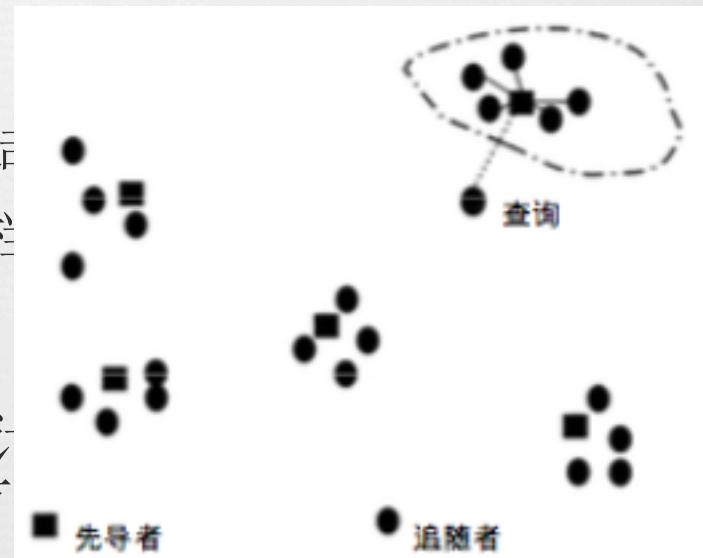
❖ 只考虑包含多个查询词项的文档

❖ 胜者表

❖ 对于词典中的每个词项 t , 预先计算所有包含 t 的文档, 给定查询 q , 对 q 中所有

❖ 簇剪枝

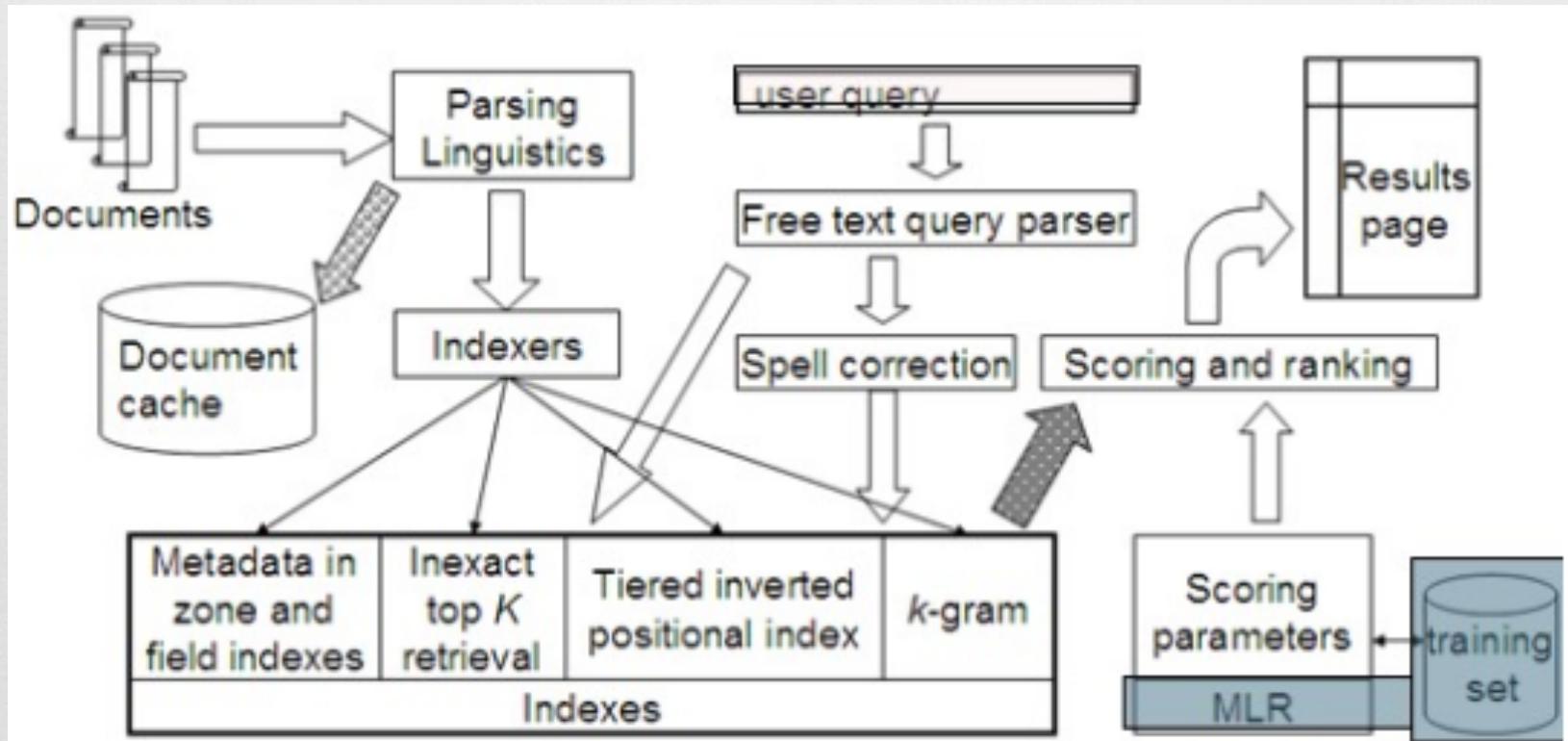
❖ 以 $\text{sqrt}(N)$ 个先导者进行聚类, 只对先导者计算相似度



信息检索系统的评价



完整的信息检索系统



查准率和查全率



- ❖ 查准率
- ❖ 查全率

$$\text{Precision} = \frac{\text{返回结果中相关文档的数目}}{\text{返回结果的数目}} = P(\text{relevant} \mid \text{retrieved})$$

$$\text{Recall} = \frac{\text{返回结果中相关文档的数目}}{\text{所有相关文档的数目}} = P(\text{retrieved} \mid \text{relevant})$$

	相关(relevant)	不相关(nonrelevant)
返回(retrieved)	真正例(true positives, tp)	伪正例(false positives, fp)
未返回(not retrieved)	伪反例(false negatives, fn)	真反例(true negatives, tn)

$$P = tp / (tp + fp)$$

$$R = tp / (tp + fn)$$

F值

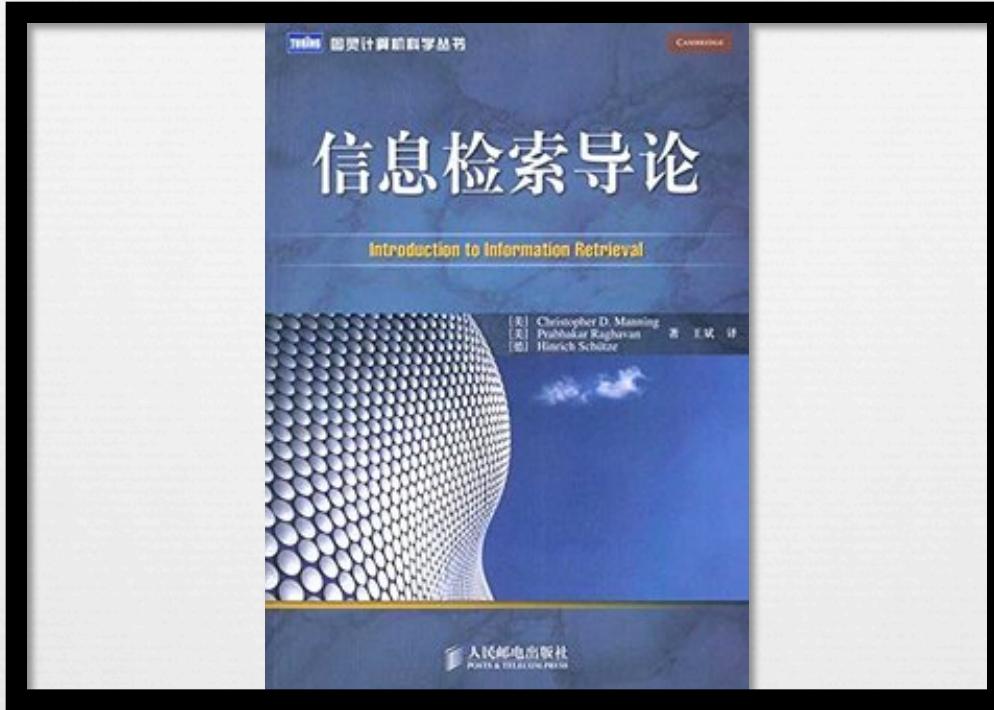


❖ 查准率和查全率的调和平均

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

❖ 特别的，当 β 为1时

$$F_{\beta=1} = \frac{2PR}{P + R}$$



谢谢！



参考文献: **Christopher D. Manning**, Prabhakar Raghavan, Hinrich Schütze. 王斌 译. 信息检索导论[M]. 北京. 人民邮电出版社. 2010 年