# 《信息检索导论》课后练习答案

王斌

最后更新日期 2013/9/28

# 第一章 布尔检索

**习题1-1 [\*]**画出下列文档集所对应的倒排索引（参考图1-3中的例子）。

文档 1 new home sales top forecasts

文档 2 home sales rise in july

文档 3 increase in home sales in july

文档 4 july new home sales rise

**解答：**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| forecasts | -------> | 1 |  |  |  |
| home | -------> | 1 🡪 | 2 🡪 | 3 🡪 | 4 |
| in | -------> | 2 🡪 | 3 |  |  |
| increase | -------> | 3 |  |  |  |
| july | -------> | 2 🡪 | 3 🡪 | 4 |  |
| new | -------> | 1 🡪 | 4 |  |  |
| rise | -------> | 2 🡪 | 4 |  |  |
| sales | -------> | 1 🡪 | 2 🡪 | 3 🡪 | 4 |
| top | -------> | 1 |  |  |  |

**习题1-2** [\*]　考虑如下几篇文档：

文档1 breakthrough drug for schizophrenia

文档2 new schizophrenia drug

文档3 new approach for treatment of schizophrenia

文档4 new hopes for schizophrenia patients

a. 画出文档集对应的词项—文档矩阵；

**解答：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 文档1 | 文档2 | 文档3 | 文档4 |
| approach | 0 | 0 | 1 | 0 |
| breakthrough | 1 | 0 | 0 | 0 |
| drug | 1 | 1 | 0 | 0 |
| for | 1 | 0 | 1 | 1 |
| hopes | 0 | 0 | 0 | 1 |
| new | 0 | 1 | 1 | 1 |
| of | 0 | 0 | 1 | 0 |
| patients | 0 | 0 | 0 | 1 |
| schizophrenia | 1 | 1 | 1 | 1 |
| treatment | 0 | 0 | 1 | 0 |

b. 画出该文档集的倒排索引（参考图 1-3中的例子）。

解答：参考a。

**习题1-3 [\*]**对于习题1-2中的文档集，如果给定如下查询，那么返回的结果是什么？

1. schizophrenia AND drug

**解答：**{文档1，文档2}

1. for AND NOT (drug OR approach）

**解答：**{文档4}

**习题1-4** [\*] 对于如下查询，能否仍然在O(x+y)次内完成？其中x和y分别是Brutus和Caesar所对应的倒排记录表长度。如果不能的话，那么我们能达到的时间复杂度是多少？

1. Brutus AND NOT Caesar
2. Brutus OR NOT Caesar

**解答：**

1. 可以在O(x+y)次内完成。通过集合的减操作即可。具体做法参考习题1-11。
2. 不能。不可以在O(x+y)次内完成。因为NOT Caesar的倒排记录表需要提取其他所有词项对应的倒排记录表。所以需要遍历几乎全体倒排记录表，于是时间复杂度即为所有倒排记录表的长度的和N，即O(N) 或者说O(x+N-y)。

**习题1-5 [\*]** 将倒排记录表合并算法推广到任意布尔查询表达式，其时间复杂度是多少？比如，对于查询

1. (Brutus OR Caesar) AND NOT (Antony OR Cleopatra)

我们能在线性时间内完成合并吗？这里的线性是针对什么来说的？我们还能对此加以改进吗？

**解答：**时间复杂度为O(qN)，其中q为表达式中词项的个数，N为所有倒排记录表长度之和。也就是说可以在词项个数q及所有倒排记录表长度N的线性时间内完成合并。由于任意布尔表达式处理算法复杂度的上界为O(N)，所以上述复杂度无法进一步改进。

**习题1-6 [\*\*]**假定我们使用分配律来改写有关AND和OR的查询表达式。

a. 通过分配律将习题1-5中的查询写成析取范式；

12

b. 改写之后的查询的处理过程比原始查询处理过程的效率高还是低？

c. 上述结果对任何查询通用还是依赖于文档集的内容和词本身？

解答：

a. 析取范式为：(Brutus And Not Anthony And Not Cleopatra) OR (Caesar AND NOT Anthony AND NOT Cleopatra)

b. 这里的析取范式处理比前面的合取范式更有效。这是因为这里先进行AND操作(括号内)，得到的倒排记录表都不大，再进行OR操作效率就不会很低。而前面需要先进行OR操作，得到的中间倒排记录表会更大一些。

c. 上述结果不一定对，比如两个罕见词A和B构成的查询 (A OR B) AND NOT(HONG OR KONG)，假设HONG KONG一起出现很频繁。此时合取方式可能处理起来更高效。如果在析取范式中仅有词项的非操作时，b中结果

不对。

**习题 1-7 [\*]**请推荐如下查询的处理次序。

1. (tangerine OR trees) AND (marmalade OR skies) AND (kaleidoscope OR eyes）

其中，每个词项对应的倒排记录表的长度分别如下：

词项 倒排记录表长度

eyes 213 312

kaleidoscope 87 009

marmalade 107 913

skies 271 658

tangerine 46 653

trees 316 812

解答：

由于：

(tangerine OR trees) 🡺 46653+316812 = 363465

(marmalade OR skies)🡺 107913+271658 = 379571

(kaleidoscope OR eyes）🡺 87009+213312 = 30321

所以推荐处理次序为：

(kaleidoscope OR eyes）AND (tangerine OR trees) AND (marmalade OR skies)

**习题1-8[\*]** 对于查询

1. friends AND romans AND (NOT countrymen)

如何利用countrymen的文档频率来估计最佳的查询处理次序？特别地，提出一种在确定查询顺序时对逻辑非进行处理的方法。

解答：令friends、romans和countrymen的文档频率分别为x、y、z。如果z极高，则将N-z作为NOT countrymen的长度估计值，然后按照x、y、N-z从小到大合并。如果z极低，则按照x、y、z从小到大合并。

**习题 1-9 [\*\*]**对于逻辑与构成的查询，按照倒排记录表从小到大的处理次序是不是一定是最优的？如果是，请给出解释；如果不是，请给出反例。

解答：不一定。比如三个长度分别为x,y,z的倒排记录表进行合并，其中x>y>z，如果x和y的交集为空集，那么有可能先合并x、y效率更高。

习题 **1-10** [\*\*]　对于查询*x* OR *y*，按照图1-6的方式，给出一个合并算法。

解答：

1 answer<- ( )

2 while p1!=NIL and p2!=NIL

3 do if docID(p1)=docID(p2)

4 then ADD(answer,docID(p1))

5 p1<- next(p1)

6 p2<-next(p2)

7 else if docID(p1)<docID(p2)

8 then ADD(answer,docID(p1))

9 p1<- next(p1)

10 else ADD(answer,docID(p2))

11 p2<-next(p2)

12 if p1!=NIL // x还有剩余

13 then while p1!=NIL do ADD (answer, docID(p1))

14 else while p2!=NIL do ADD(answer,docID(p2))

15 return(answer)

习题 1-11 [\*] 如何处理查询x AND NOT y？为什么原始的处理方法非常耗时？给出一个针对该查询的高效合并算法。

解答：由于NOT y几乎要遍历所有倒排表，因此如果采用列举倒排表的方式非常耗时。可以采用两个有序集合求减的方式处理 x AND NOT y。算法如下：

Meger(p1,p2)

1 answer ()

2 while p1!=NIL and p2!=NIL

3 do if docID(p1) =docID(p2)

4 then p1🡨next(p1)

5 p2🡨next(p2)

6 else if docID(p1)<docID(p2)

7 then ADD(answer, docID(p1))

8 p1🡨next(p1)

9 else ADD(answer, docID(p2))

10 p2🡨next(p2)

11 if p1!=NIL // x还有剩余

12 then while p1!=NIL do ADD (answer, docID(p1))

13 return(answer)

习题 **1-12** [\*]　利用Westlaw系统的语法构造一个查询，通过它可以找到professor、teacher或lecturer中的任意一个词，并且该词和动词explain在一个句子中出现，其中explain以某种形式出现。

解答： professor teacher lecturer /s explain!

习题 **1-13** [\*]　在一些商用搜索引擎上试用布尔查询，比如，选择一个词（如burglar），然后将如下查询提交给搜索引擎

(i) burglar；(ii)burglar AND burglar；(iii) burglar OR burglar。

对照搜索引擎返回的总数和排名靠前的文档，这些结果是否满足布尔逻辑的意义？对于大多数搜索引擎来说，它们往往不满足。你明白这是为什么吗？如果采用其他词语，结论又如何？比如以下查询

(i) knight；(ii) conquer；(iii) knight OR conquer。

# 第二章 词汇表和倒排记录表

习题 **2-1** [*\**]　请判断如下说法是否正确。

a. 在布尔检索系统中，进行词干还原从不降低正确率。

b. 在布尔检索系统中，进行词干还原从不降低召回率。

c. 词干还原会增加词项词典的大小。

d. 词干还原应该在构建索引时调用，而不应在查询处理时调用。

解答： a错 b 对 c错 d 错

习题**2-7** [\*]　考虑利用如下带有跳表指针的倒排记录表

36

和一个中间结果表（如下所示，不存在跳表指针）进行合并操作。

3　5　89　95　97　99　100　101

采用图2-10所示的倒排记录表合并算法，请问：

1. 跳表指针实际跳转的次数是多少（也就是说，指针*p*1的下一步将跳到skip（p1））？

一次，24—>75

1. 当两个表进行合并时，倒排记录之间的比较次数是多少？【如下答案不一定正确，有人利用程序计算需要21次，需要回到算法，本小题不扣分，下面不考虑重新比较同意对数字】

解答：  18次： <3,3>, <5,5>, <9,89>, <15,89>,<24,89>,<75,89>,<92,89>,<81,89>,<84,89>,<89,89>,<92,95>,<115,95>,<96,95>,<96,97>,<97,97>,<100,99>,<100,100><115,101>

1. 如果不使用跳表指针，那么倒排记录之间的比较次数是多少？

解答： 19次： <3,3>,<5,5>,<9,89>,<15,89>,<24,89>,<39,89>,<60,89>,<68,89>,<75,89>,<81,89>,<84,89>,<89,89><92,95>, <96,95>,<96,97>,<97,97>,<100,99>,<100,100>,<115,101>

习题 **2-9** [*\**]　下面给出的是一个位置索引的一部分，格式为：词项: 文档1: 〈位置1, 位置2, …〉; 文档2: 〈位置1, 位置2, …〉。

angels: 2: 〈36,174,252,651〉; 4: 〈12,22,102,432〉; 7: 〈17〉;

fools: 2: 〈1,17,74,222〉; 4: 〈8,78,108,458〉; 7: 〈3,13,23,193〉;

fear: 2: 〈87,704,722,901〉; 4: 〈13,43,113,433〉; 7: 〈18,328,528〉;

in: 2: 〈3,37,76,444,851〉; 4: 〈10,20,110,470,500〉; 7: 〈5,15,25,195〉;

rush: 2: 〈2,66,194,321,702〉; 4: 〈9,69,149,429,569〉; 7: 〈4,14,404〉;

to: 2: 〈47,86,234,999〉; 4: 〈14,24,774,944〉; 7: 〈199,319,599,709〉;

tread: 2: 〈57,94,333〉; 4: 〈15,35,155〉; 7: 〈20,320〉;

where: 2: 〈67,124,393,1001〉; 4: 〈11,41,101,421,431〉; 7: 〈16,36,736〉;

那么哪些文档和以下的查询匹配？其中引号内的每个表达式都是一个短语查询。

1. “fools rush in”。

解答： 文档2、4、7

b. “fools rush in” AND “angels fear to tread”。

解答： 文档4

# 第三章 词典及容错式检索

习题 **3-5**　再次考虑3.2.1节中的查询fi\*mo\*er，如果采用2-gram索引的话，那么对应该查询应该会产生什么样的布尔查询？你能否举一个词项的例子，使该词匹配3.2.1节的轮排索引查询，但是并不满足刚才产生的布尔查询？

解答： 2-gram索引下的布尔查询：$f AND fi AND mo AND er AND r$

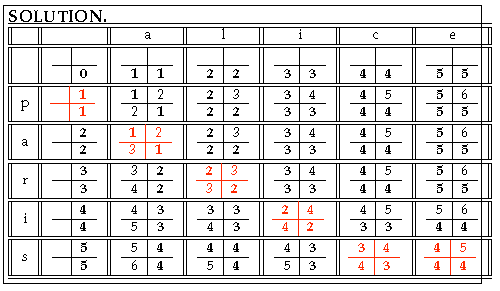
词项filibuster(海盗)满足 3.2.1节的轮排索引查询，但是并不满足上述布尔查询

习题 3-7　如果 |si | 表示字符串si的长度，请证明s1和s2的编辑距离不可能超过max{|s1|, |s2|}。

证明：不失一般性，假设|s1|<= |s2|，将s1转换为s2的一种做法为：将s1中的每个字符依次替换为s2中的前|s1|个字符，然后添加s2的后|s2|-|s1|个字符，上述操作的总次数为|s2|= max{|s1|, |s2|}，根据编辑距离的定义，其应该小于|s2|= max{|s1|, |s2|}

习题 **3-8**计算paris和 alice之间的编辑距离，给出类似于图3-5中的算法结果，其中的5 × 5 矩阵包含每个前缀子串之间的计算结果。

解答：



习题 **3-11**考虑四词查询catched in the rye，假定根据独立的词项拼写校正方法，每个词都有5个可选的正确拼写形式。那么，如果不对空间进行缩减的话，需要考虑多少可能的短语拼写形式（提示：同时要考虑原始查询本身，也就是每个词项有6种变化可能）？

57

解答：6\*6\*6\*6=1296

习题 **3-14**找出两个拼写不一致但soundex编码一致的专有名词。

解答：Mary, Mira (soundex相同)，本题答案不唯一，可能有其他答案，但是soundex编码必须一致。

# 第四章 索引构建

未标题-1习题 **4-1**　如果需要*T* log2 *T*次比较（T是词项ID—文档ID对的数目），每次比较都有两次磁盘寻道过程。假定使用磁盘而不是内存进行存储，并且不采用优化的排序算法（也就是说不使用前面提到的外部排序算法），那么对于Reuters-RCV1构建索引需要多长时间？计算时假定采用表4-1中的系统参数。

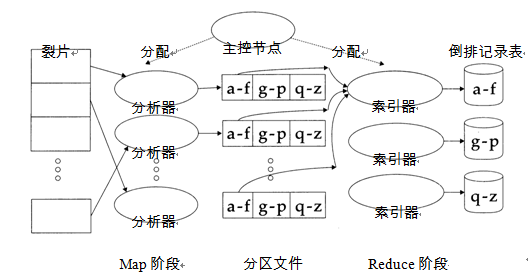
解答：

对于Reuters-RCV1，T=108

因此排序时间(文档分析时间可以忽略不计)为：2\*(108\*log2108)\*5\*10-3s = 26575424s = 7382 h=308 day

习题 **4-3**对于*n* = 15个数据片，*r* = 10个分区文件，*j* = 3个词项分区，假定使用的集群的机器的参数如表4-1所示，那么在MapReduce构架下对Reuters-RCV1语料进行分布式索引需要多长时间？

【给助教：教材不同印刷版本表4-2不一样，不同同学用的不同版本，还有本题过程具有争议。暂不扣分】



解答【整个计算过程是近似的，要了解过程】：

**(一)、MAP阶段**【读入语料(已经不带XML标记信息了，参考表5-6)，词条化，写入分区文件】：

(1) 读入语料：

基于表4-2，Reuters RCV1共有8\*105篇文档，每篇文档有200词条，每个词条(考虑标点和空格)占6B，因此整个语料库的大小为 8\*105\*200\*6=9.6\*108B （近似1GB，注表4-2对应于表5-1第3行的数据，而那里的数据已经经过 去数字 处理，因此实际的原始文档集大小应该略高于0.96G，这里近似计算，但是不要认为没有处理就得到表5-1第3行的结果）

将整个语料库分成15份，则每份大小为9.6\*108/15 B

每一份读入机器的时间为：9.6\*108/15\*2\*10-8=1.28s

(2) 词条化：每一份语料在机器上进行词条化处理，得到8\*105\*200=1.6\*108个词项ID-文档ID对(参考表4-2和图4-6，注意此时重复的 词项ID-文档ID对 还没有处理)，共占1.6\*108\*8=1.28\*109个字节，词条化的时间暂时忽略不计【从题目无法得到词条化这一部分时间，从表5-1看词条化主要是做了去数字和大小写转换，当然也感觉这一部分的处理比较简单，可以忽略】。

(3) 写入分区文件：每一份语料得到的词项ID-文档ID (Key-Value)存储到分区所花的时间为：

(1.28\*109/15)\*2\*10-8=1.71s

(4) MAP阶段时间：

由于分成15份，但只有10台机器进行MAP操作，所以上述MAP操作需要两步，因此，整个MAP过程所需时间为 (1.28+1.71)\*2=6.0s

**(二)、REDUCE阶段**【读入分区文件，排序，写入倒排索引】：

(1) 读入分区文件【读入过程中已经实现所有Key-Value对中的Value按Key聚合，即变成Key, list(V1,V2..)。聚合过程在内存中实现，速度很快，该时间不计。另外，网络传输时间这里也不计算】：

根据表4-2，所有倒排记录的数目为1.6\*108，因此3台索引器上每台所分配的倒排记录数目为 1.6\*108/3,而每条记录由4字节词项ID和4字节文档ID组成，因此每台索引器上需要读入的倒排记录表数据为 1.28\*109/3字节。

于是，每台索引器读数据的时间为 1.28\*109/3\*2\*10-8=8.5s

(2) 排序：

每台索引器排序所花的时间为 1.6\*108/3\*log2(1.6\*108/3)\*10-8=13.7s

(3) 写入倒排索引文件【此时倒排文件已经实现文档ID的去重，假定只存储词项ID和文档ID列表，并不存储其他信息(如词项的DF及在每篇文档中的TF还有指针等等)】：

需要写入磁盘的索引大小为(据表4-2，词项总数为4\*105个) 4\*105/3\*4+108/3\*4=4/3\*108字节

索引写入磁盘的时间为：4/3\*108\*2\*10-8=2.7s

(4) REDUCE阶段时间为： 8.5+13.7+2.7=24.9

(三) 因此，整个分布式索引的时间约为6.0+8.5+13.7+2.7=30.9s

# 第五章 索引压缩

习题 **5-2**估计Reuters-RCV1文档集词典在两种不同按块存储压缩方法下的空间大小。其中，第一种方法中*k* = 8，第二种方法中*k* = 16。

解答：

每8个词项会节省7\*3个字节，同时增加8个字节，于是每8个词项节省7\*3-8=13字节，所有词项共节省13\*400000/8=650K，因此，此时索引大小为7.6MB-0.65MB=6.95MB

每16个词项会节省15\*3个字节，同时增加16个字节，于是每16个词项节省15\*3-16=29字节，所有词项共节省29\*400000/16=725K，因此，此时索引大小为7.6MB-0.725MB=6. 875MB

习题 **5-6**考虑倒排记录表**（**4*,* 10*,* 11*,* 12*,* 15*,* 62*,* 63*,* 265*,* 268*,* 270*,* 400）及其对应的间距表（4*,* 6*,* 1*,* 1*,* 3*,* 47*,* 1*,* 202*,*3*,* 2*,* 130）。假定倒排记录表的长度和倒排记录表分开独立存储，这样系统能够知道倒排记录表什么时候结束。采用可变字节码：

(i) 能够使用1字节来编码的最大间距是多少？

(ii) 能够使用2字节来编码的最大间距是多少？

(iii) 采用可变字节编码时，上述倒排记录表总共需要多少空间（只计算对这些数字序列进行编码的空间消耗）？

解答：

(i) 27-1=127 (答128也算对，因为不存在0间距，0即可表示间距1，……)

(ii) 214-1=16383 (答16384也算对)

(iii) 1+1+1+1+1+1+1+2+1+1+2=13

习题 **5-8 [**\***]**　对于下列采用*γ* 编码的间距编码结果，请还原原始的间距序列及倒排记录表。

1110001110101011111101101111011

解答：

1110 001; 110 10; 10 1; 111110 11011; 110 11

1001; 110; 11; 111011; 111

9; 6; 3; 32+16+8+2+1=59; 7

9; 15;18;77;84

# 第六章 文档评分、词项权重计算及向量空间模型

习题 **6-10**考虑图6-9中的3篇文档Doc1、Doc2、Doc3中几个词项的tf情况，采用图6-8中的idf值来计算所有词项car、auto、insurance及best的tf-idf值。

|  |  |  |  |
| --- | --- | --- | --- |
|  | Doc1 | Doc2 | Doc3 |
| car | 27 | 4 | 24 |
| auto | 3 | 33 | 0 |
| insurance | 0 | 33 | 29  **3** |
| best | 14 | 0 | 17 |

图6-9　习题 6-10中所使用的tf值

解答：

idfcar=1.65，idfauto=2.08，idfinsurance=1.62，idfbest=1.5，

于是，各词项在各文档中的tf-idf结果如下表：

|  |  |  |  |
| --- | --- | --- | --- |
|  | Doc1 | Doc2 | Doc3 |
| car | 27\*1.65=44.55 | 4\*1.65=6.6 | 24\*1.65=39.6 |
| auto | 3\*2.08=6.24 | 33\*2.08=68.64 | 0 |
| insurance | 0 | 33\*1.62=53.46 | 29\*1.62=46.98  **3** |
| best | 14\*1.5=21 | 0 | 17\*1.5=25.5 |

习题 **6-12**公式（6-7）中对数的底对公式（6-9）会有什么影响？对于给定查询来说，对数的底是否会对文档的排序造成影响？

解答：没有影响。

假定idf采用与(6-7)不同的底x计算，根据对数换底公式有。

idft(x)=logx(N/dft)=log(N/dft)/logx=idft/logx，

由于idft(x)和idft之间只相差一个常数因子1/logx，在公式(6-9)的计算中该常数可以作为公因子提出，因此文档的排序不会改变。

习题 **6-19**计算查询digital cameras及文档digital cameras and video cameras的向量空间相似度并将结果填入表6-1的空列中。假定*N*=10 000 000，对查询及文档中的词项权重（wf对应的列）采用对数方法计算，查询的权重计算采用idf，而文档归一化采用余弦相似度计算。将 and 看成是停用词。请在tf列中给出词项的出现频率，并计算出最后的相似度结果。

121

表6-1　习题6-19中的余弦相似度计算

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 词 | 查　　询 | | | | | **文　　档** | | | |  |
| tf | wf | df | idf | *qi*=wf-idf | tf | wf | *di*=归一化的wf | |
| digital |  |  | 10 000 |  |  |  |  |  |  |  |
| video |  |  | 100 000 |  |  |  |  |  |  |  |
| cameras |  |  | 50 000 |  |  |  |  |  |  |  |

解答：【本质上这里没有考虑查询向量的归一化，即没有考虑查询向量的大小，严格上不是余弦相似度】

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 词 | 查　　询 | | | | | **文　　档** | | | |  |
| tf | wf | df | idf | *qi*=wf-idf | tf | wf | *di*=归一化的wf | |
| digital | 1 | 1 | 10 000 | 3 | 3 | 1 | 1 | 0.520 |  |  |
| video | 0 | 0 | 100 000 | 2 | 0 | 1 | 1 | 0.520 |  | 3.112 |
| cameras | 1 | 1 | 50 000 | 2.301 | 2.301 | 2 | 1.301 | 0.677 |  |  |

习题 **6-23**　考虑习题 6-10中4个词项和3篇文档中的tf和idf值，采用如下权重计算机制来计算获得得分最高的两篇文档：(i) nnn.atc ；(ii) ntc.atc。

解答：(i) 根据题意文档采用nnn，查询采用atc，然后计算内积，于是有：

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 词项 | 查询q | | | | 文档Doc1 | | | 得分 |
| tf | idf | tf-idf | 归一化tf-idf | tf | idf | tf-idf |
| car | 1 | 1.65 | 1.65 | 0.560 | 27 | 1 | 27 | 23.310 |
| auto | 0.5 | 2.08 | 1.04 | 0.353 | 3 | 1 | 3 |
| insurance | 1 | 1.62 | 1.62 | 0.550 | 0 | 1 | 0 |
| best | 1 | 1.5 | 1.5 | 0.509 | 14 | 1 | 14 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 词项 | 查询q | | | | 文档Doc2 | | | 得分 |
| tf | idf | tf-idf | 归一化tf-idf | tf | idf | tf-idf |
| car | 1 | 1.65 | 1.65 | 0.560 | 4 | 1 | 4 | 32.037 |
| auto | 0.5 | 2.08 | 1.04 | 0.353 | 33 | 1 | 33 |
| insurance | 1 | 1.62 | 1.62 | 0.550 | 33 | 1 | 33 |
| best | 1 | 1.5 | 1.5 | 0.509 | 0 | 1 | 0 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 词项 | 查询q | | | | 文档Doc3 | | | 得分 |
| tf | idf | tf-idf | 归一化tf-idf | tf | idf | tf-idf |
| car | 1 | 1.65 | 1.65 | 0.560 | 24 | 1 | 24 | 38.046 |
| auto | 0.5 | 2.08 | 1.04 | 0.353 | 0 | 1 | 0 |
| insurance | 1 | 1.62 | 1.62 | 0.550 | 29 | 1 | 29 |
| best | 1 | 1.5 | 1.5 | 0.509 | 17 | 1 | 17 |

      于是，在nnn.atc下，Score(q,Doc3)>Score(q,Doc2)>Score(q,Doc1)

(ii) 根据题意文档采用ntc，查询采用atc，然后计算内积，于是有：

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 词项 | 查询q | | | | 文档Doc1 | | | | 得分 |
| tf(a) | idf | tf-idf | 归一化tf-idf | tf | idf | tf-idf | 归一化  tf-idf |
| car | 1 | 1.65 | 1.65 | 0.560 | 27 | 1.65 | 44.55 | 0.897 | 0.76 |
| auto | 0.5 | 2.08 | 1.04 | 0.353 | 3 | 2.08 | 6.24 | 0.125 |
| insurance | 1 | 1.62 | 1.62 | 0.550 | 0 | 1.62 | 0 | 0 |
| best | 1 | 1.5 | 1.5 | 0.509 | 14 | 1.5 | 21 | 0.423 |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 词项 | 查询q | | | | 文档Doc2 | | | | 得分 |
| tf(a) | idf | tf-idf | 归一化tf-idf | tf | idf | tf-idf | 归一化  tf-idf |
| car | 1 | 1.65 | 1.65 | 0.560 | 4 | 1.65 | 6.6 | 0.075 | 0.66 |
| auto | 0.5 | 2.08 | 1.04 | 0.353 | 33 | 2.08 | 68.64 | 0.786 |
| insurance | 1 | 1.62 | 1.62 | 0.550 | 33 | 1.62 | 53.46 | 0.613 |
| best | 1 | 1.5 | 1.5 | 0.509 | 0 | 1.5 | 0 | 0 |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 词项 | 查询q | | | | 文档Doc3 | | | | 得分 |
| tf(a) | idf | tf-idf | 归一化tf-idf | tf | idf | tf-idf | 归一化  tf-idf |
| car | 1 | 1.65 | 1.65 | 0.560 | 24 | 1.65 | 39.6 | 0.595 | 0.92 |
| auto | 0.5 | 2.08 | 1.04 | 0.353 | 0 | 2.08 | 0 | 0 |
| insurance | 1 | 1.62 | 1.62 | 0.550 | 29 | 1.62 | 46.98 | 0.706 |
| best | 1 | 1.5 | 1.5 | 0.509 | 17 | 1.5 | 25.5 | 0.383 |

      于是，在nnn.atc下，Score(q,Doc3)>Score(q,Doc1)>Score(q,Doc2)

# 第七章 一个完整搜索系统中的评分计算

习题 **7-3**　给定单个词项组成的查询，请解释为什么采用全局胜者表（*r*=*K*）已经能够充分保证找到前*K*篇文档。如果只有*s*个词项组成的查询（*s*>1），如何对上述思路进行修正？

解答：

词项t所对应的tf最高的r篇文档构成t的胜者表。单词项查询，idf已经不起作用了(idf用于区别不同词的先天权重)，所以此时已经足够了。

对于s个词项组成的查询，有idf权重了。。因此，不再独立。【这一问本人也不知道该怎么答，不扣分吧】

习题 **7-5**　重新考察习题6-23中基于nnn.atc权重计算的数据，假定Doc1和Doc2的静态得分分别是1和2。请确定在公式（7-2）下，如何对Doc3的静态得分进行取值，才能分别保证它能够成为查询best car insurance的排名第一、第二或第三的结果。

解答：这道题不扣分吧。。整个书上有关余弦相似度的计算这块都有问题【即按照公式(7-2) (6-12)算出的应该是0到1之间的数，但实际例子(例6-4)却是大于1的数，例子中都没有考虑查询向量的大小。另外，按照习题6-23中nnn.atc算出的根本不是什么余弦相似度。整个一团乱】

如果相似度先采用nnn.atc计算，最后除以文档向量的大小，则三篇文档的得分分别为：1.39、1.47和1.68。

* + 排名第一：g(d3)+1.68>3.47, g(d3)>1.79
  + 排名第二：2.39< g(d3)+1.68 <3.47, 0.71< g(d3)<1.79
  + 排名第三：0< g(d3) < 0.71

习题 **7-7**设定图6-10中Doc1、Doc2和Doc3的静态得分分别是0.25、0.5和1，画出当使用静态得分与欧几里得归一化tf值求和结果进行排序的倒排记录表。

解答：按照公式7-2计算得下表：

|  |  |  |  |
| --- | --- | --- | --- |
|  | doc1 | doc2 | doc3 |
| car | 1.13 | 0.59 | 1.58 |
| auto | 0.35 | 1.21 | 1 (0) |
| insurance | 0.25 (0) | 1.21 | 1.7 |
| best | 0.71 | 0.5 (0) | 1.41 |

所以，倒排记录表如下：

car 🡪doc3 🡪doc1 🡪doc2

auto 🡪doc2🡪~~doc 3 🡪~~doc1 【按道理，tf为零的不应该出现在倒排记录中，有的也算对】

insurance🡪doc3🡪doc2~~🡪doc1~~

best 🡪doc3🡪doc1~~🡪doc2~~

# 第八章 信息检索的评价

习题 **8-8** [*\**]　考虑一个有4篇相关文档的信息需求，考察两个系统的前10个检索结果（左边的结果排名靠前），相关性判定的情况如下所示：

系统1 R N R N N N N N R R

系统2 N R N N R R R N N N

a. 计算两个系统的MAP值并比较大小。

b. 上述结果直观上看有意义吗？能否从中得出启发如何才能获得高的MAP得分？

c. 计算两个系统的R正确性值，并与a中按照MAP进行排序的结果进行对比。

解答：

a. 系统1 (1+2/3+3/9+4/10)/4=0.6

系统2 (1/2+2/5+3/6+4/7)/4=0.492

b. 相关文档出现得越靠前越好，最好前面3-5篇之内

c. 系统1的R-Precision= 0.5, 系统2 R-Precision= 0.25

习题 **8-9** [*\*\**]　在10 000篇文档构成的文档集中，某个查询的相关文档总数为8，下面给出了某系统针对该查询的前20个有序结果的相关（用R表示）和不相关（用N表示）情况，其中有6篇相关文档：

R R N N N N N N R N R N N N R N N N N R

1. 前20篇文档的正确率是多少？

P@20=6/20=30%

1. 前20篇文档的*F*1值是多少?

R@20=6/8=75%，F1=3/7=0.429

1. 在25%召回率水平上的插值正确率是多少？

150

1

1. 在33%召回率水平上的插值正确率是多少？

3/9=33.3%

1. 假定该系统所有返回的结果数目就是20，请计算其MAP值。

(1+1+3/9+4/11+5/15+6/20)/8=0.4163

假定该系统返回了所有的10 000篇文档，上述20篇文档只是结果中最靠前的20篇文档，那么

1. 该系统可能的最大MAP是多少？

从第21位开始，接连两篇相关文档，此时可以获得最大的MAP，此时有：

(1+1+3/9+4/11+5/15+6/20+7/21+8/22)/8=0.503

1. 该系统可能的最小MAP是多少？

(1+1+3/9+4/11+5/15+6/20+7/9999+8/10000)/8=0.4165

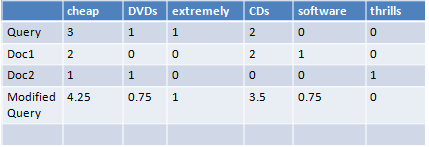
h. 在一系列实验中，只有最靠前的20篇文档通过人工来判定，(e)的结果用于近似从(f)到(g)的MAP取值范围。对于上例来说，通过(e)而不是(f)和(g)来计算MAP所造成的误差有多大（采用绝对值来计算）？

|0.4163-(0.503+0.4165)/2|=0.043

# 第九章 相关反馈及查询扩展

习题9-3：用户查看了两篇文档d1 和 d2，并对这两篇文档进行了判断：包含内容CDs cheap software cheap CDs的文档d1为相关文档，而内容为cheap thrills DVDs 的文档d2为不相关文档。假设直接使用词项的频率作为权重 （不进行归一化也不加上文档频率因子），也不对向量进行长度归一化。采用公式（9-3）进行Rocchio相关反馈，请问修改后的查询向量是多少？其中α = 1，β = 0.75，γ = 0.25。

解答：

****

习题9-4： Omar实现了一个带相关反馈的Web搜索系统，并且为了提高效率，系统只基于返回网页的标题文本进行相关反馈。用户对结果进行判定，假定第一个用户Jinxing的查询是

banana slug

返回的前三个网页的标题分别是：

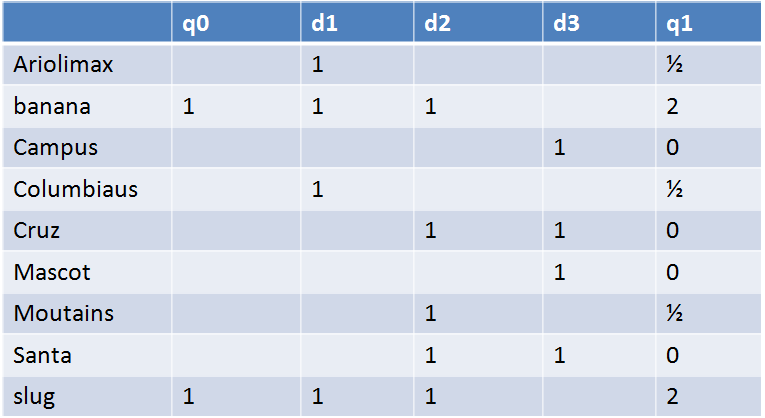
banana slug Ariolimax columbianus

Santa Cruz mountains banana slug

Santa Cruz Campus Mascot

Jinxing认为前两篇文档相关，而第3篇文档不相关。假定Omar的搜索引擎只基于词项频率（不包括长度归一化因子和IDF因子）进行权重计算，并且假定使用Rocchio算法对原始查询进行修改，其中α = β = γ = 1。请给出最终的查询向量（按照字母顺序依次列出每个词项所对应的分量）。

解答：



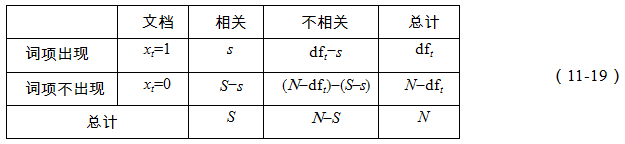
# 第十章 XML检索

**(无作业)**

# 第十一章 概率检索模型

习题11-1 根据公式（11-18）和公式（11-19）推导出公式（11-20）。

****

****

****

解答：代入求解即可。

习题11-3 令Xt表示词项t在文档中出现与否的随机变量。假定文档集中有|R|篇相关文档，其中有s篇文档包含词项t，即在这s篇文档中Xt=1。假定所观察到的数据就是这些Xt在文档中的分布情况。请证明采用MLE估计方法对参数进行估计的结果，即使得观察数据概率最大化的参数值为 pt = s/ |R|。

# 第十二章 基于语言建模的信息检索模型

习题12-3 习题12-3 例12-2中按照M1 和 M2 算出的文档的似然比是多少？

解答：由于P(s|M1) = 0.000 000 000 000 48

，P(s|M2) = 0.000 000 000 000 000 384，所以两者的似然比是 0.00000000000048/ 0.000000000000000384 =1250

习题12-6 [\*]　考虑从如下训练文本中构造LM：

the martian has landed on the latin pop sensation ricky martin

请问：

a. 在采用MLE估计的一元概率模型中，P(the)和P(martian)分别是多少？

b. 在采用MLE估计的二元概率模型中，P(sensation|pop)和 P(pop|the)的概率是多少？

解答：

a. P(the)=2/11, P(martian)=1/11

b. P(sensation|pop)=1, P(pop|the)=0

习题 12-7 [\*\*]　假定某文档集由如下4篇文档组成：

|  |  |
| --- | --- |
| **文档ID** | **文档文本** |
| **1**  **2**  **3**  **4** | **click go the shears boys click click click**  **click click**  **metal here**  **metal shears click here** |

为该文档集建立一个查询似然模型。假定采用文档语言模型和文档集语言模型的混合模型，权重均为0.5。采用MLE来估计两个一元模型。 计算在查询click、shears以及click shears下每篇文档模型对应的概率，并利用这些概率来对返回的文档排序。将这些概率填在下表中。

解答：

文档及文档集MLE估计

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **query** | **doc1** | **doc2** | **doc3** | **doc4** | **collection** |
| **click** | **1/2** | **1** | **0** | **1/4** | **7/16** |
| **shears** | **1/8** | **0** | **0** | **1/4** | **2/16** |

于是，加权以后的估计结果 doc4> doc1>doc2>doc3

# 第十三章 文本分类及朴素贝叶斯方法

习题 **13-2 [**\***]**　表13-5中的文档中，对于如下的两种模型表示，哪些文档具有相同的模型表示？哪些文档具有不同的模型表示？对于不同的表示进行描述。(i) 贝努利模型，(ii) 多项式模型。

习题13-9

# 第十四章 基于向量空间模型的文本分类

# 第十五章 支持向量机及文档机器学习方法

# 第十六章 扁平聚类

# 第十七章 层次聚类

# 第十八章 矩阵分解及隐性语义索引

# 第十九章 Web搜索基础

# 第二十章 Web采集及索引

# 第二十一章 链接分析