《信息组织与检索》作业答案

第一章　布尔检索

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；

b. 画出该文档集的倒排索引（参考图　　1-3 中的例子）。

Term-Documentmatrix:

	1	2	3	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

Inverted Index:

approach -> 3

breakthrough ->1

drug ->1->2

for ->1->3->4

hopes ->4

new ->2->3->4

of ->3

patients ->4

schizophrenia ->1->2->3->4

treatment >3

注意：倒排索引中的词表（　dictionary）和每个词项的倒排列表（　posting list）需要排序，便

于查找。这里我们暂不考虑词的正规化处理（如 hopes->hope）。

补充习题 1

写出 AND 查询的伪代码

面向过程风格的伪代码：

给定两个指针 p1 和 p2，分别指向两倒排列表 list1 和 list2（链表实现） 的首元素； 令 docId(p1) 表示 p1 所指向的元素的 docId 查询结果存放在 answer 列表里。

这里应用了“ 化归 ”思想（将新问题转化归为旧问题来解决） 。这里，比较两排序列表的首元素，排除较小的 docId（不可能有匹配）后，我们构造出新的剩余列表，再次进行两列表的首元素的比较。

```
While p1 != null AND p2 != null
   If p1->docId==p2->docId //  对两（剩余）列表的首元素进行比较
      insert(answer, p1) ；
      p1=p1->next ； // 构造新的剩余列表，迭代执行
      p2=p2->next ； //
   Else if p1->docId < p2->docId
      p1=p1->next ；  //p1->docId  不可能有匹配；构造新的剩余列表
   Else
      p2=p2->next ；  //p2->docId  不可能有匹配；构造新的剩余列表
   End
```

面向对象风格的伪代码：

注：为一个数据结构（对象）定义方法，通过方法操作自己的内部数据（ List 对象里隐含包含了一个成员变量，它是真正的链表或变长数组） 。

```
While list1.currentItem() != null AND list2.currentItem() != null
      If list1.currentItem().getDocId() == list2.currentItem().getDocId()
      answer.insert(list1.currentItem());
      list1.moveToNext();
      list2.moveToNext();
Else if list1.currentItem().getDocId() < list2.currentItem().getDocId()
      list1.moveToNext();
Else
      list2.moveToNext();
End
```

习题 **1-10**

写出 OR 查询的伪代码

面向过程风格的伪代码：

给定两个指针 p1 和 p2, 分别指向两倒排列表 list1 和 list2（链表实现） 的首元素； 令 docId(p1) 表示 p1 所指向的元素的 docId；查询结果存放在 answer 列表里。

```
While p1 != null  AND p2 != null
    If p1->docId == p2->docId
        insert(answer, p1) ；
        p1=p1->next ；
        p2=p2->next ；     // 构造新的剩余列表，迭代执行
    Else if p1->docId < p2->docId
        insert(answer, p1) ；
        p1=p1->next ；  // 构造新的剩余列表，迭代执行
    Else
        insert(answer, p2) ；
        p2=p2->next ；  // 构造新的剩余列表，迭代执行
    End

While p1 != null//  条件为真时，加入 list1 的剩余元素（此时 list2 已遍历到结尾）
        insert(answer, p1) ；
        p1=p1->next ；
END

While p2 != null//  条件为真时，加入 list2 的剩余元素（此时 list1 已遍历到结尾）
        insert(answer, p2) ；
        p2=p1->next ；
END
```

面向对象风格的伪代码：

```
While list1.currentItem() != null AND list2.currentItem() != null
        If list1.currentItem().getDocId() == list2.currentItem().getDocId()
                answer.insert(list1.currentItem());
                list1.moveToNext();
                list2.moveToNext();
        Else if list1.currentItem().getDocId() < list2.currentItem().getDocId()
                answer.insert(list1.currentItem());
                list1.moveToNext();
```

```
        Else
            answer.insert(list2.currentItem());
            list2.moveToNext();
        End

While list1.currentItem() != null
        answer.insert(list1.currentItem());
        list1.moveToNext();
END

While list2.currentItem() != null
        answer.insert(list2.currentItem());
        list2.moveToNext();
END
```

补充习题 **2**

若一个文集有 1000 篇文档，有 40 篇是关于信管专业建设的。我的信息需求是了解信管专业的专业建设情况，用某搜索引擎在这个文集上搜索，查询词为“信管”，搜出 100 篇包含“信管”的文档，这其中有 20 篇是信管专业建设方面的，其它 80 篇是关于信管的其它情况。请问该查询的正确率和召回率是多少

正确率 =20/100=0.2

召回率 =20/40=0.5

第二章 词项词典及倒排记录表

习题 **2-1**

a. 在布尔检索系统中，进行词干还原从不降低正确率。
   错；相当于扩充出同一个词干表示的多个词，会降低正确率。

b. 在布尔检索系统中，进行词干还原从不降低召回率。
   对。

c. 词干还原会增加词项词典的大小。
   错。

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。
   错；应同时做才能保证索引中和查询词的匹配。

习题 2-2

请给出如下单词的归一化形式（归一化形式也可以是词本身）　　。

a. ’C->cos

b. Shi ’ite->shiite（'是隔音号）

c. cont ->contd（contd. 可表示 contained 包括； continued 继续）

d. Hawai ->hawaii

e. O ’Rourke->orourke

习题 2-3

如下词经过 Porter 词干还原工具处理后会输出同样的结果，你认为哪对（几对）词不应该输出同样的结果？为什么？

a. abandon/abandonment

b. absorbency/absorbent

c. marketing/markets

d. university/universe

e. volume/volumes

按 Porter 词干还原算法， 这几组词都可以被还原为相应的词干。 但是这里问的是哪些组做词干还原不合适， 原因是某组的两个词虽然来源于同一个词干， 但是它们的意思不同， 如果做词干还原处理会降低正确率。

c 组不做词干还原。 marketing 表示营销， market 表示市场。

d 组不做词干还原。 university 表示大学， universe 表示宇宙。

习题 2-6

对于两个词组成的查询，其中一个词（项）的倒排记录表包含下面 16 个文档 ID：

[4,6,10,12,14,16,18,20,22,32,47,81,120,122,157,180]

而另一个词（项）对应的倒排记录表仅仅包含一个文档 ID：

[47]

请分别采用如下两种策略进行倒排记录表合并并计算所需要的比较次数， 同时简要地说明计算的正确性。

a. 使用标准的倒排记录表。

比较： (4,47)，(6,47)，(10,47)，(12,47)，(14,47)，(16,47)，(18,47)，(20,47)，(22,47)，(32,47)，(47,47)。共比较 11 次。

b. 使用倒排记录表 +跳表的方式，跳表指针设在 $P^{1/2}$ 处（ P 是列表长度）。

P=16。也就说第一个列表的跳表指针往后跳　　4 个元素。
下图蓝色表示安装了跳表指针的元素，其中　　120 跳到 180 上。

[4,6,10,12,14,16,18,20,22,32,47,81,120,122,157,180]

比较：(4,47), (14,47), (22,47), (120,47), (32,47), (47,47) 。共比较 6 次。

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。

a. “fools rush in”；

文档 2、4、7。

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

文档 4。

补充习题 **1**

k 词邻近 **AND** 合并算法

前提：

考虑位置索引。

要求查找这样的文档，它同时包含词　　A 和词 B，且两词文中的距离在　　k 个词以内。

给定两个指针　p1 和 p2，分别指向两个词　A 和 B 的两倒排列表（链表实现）的首元素；令 pi->doc 表示 pi 所指向文档对象的结构体。

对于一个文档对象，　该词出现的各个位置的列表为　　posList。用 q1（q2）表示词 A（词 B）当前指向文档对象指向的　　posList 指向的位置。用　qi->pos 表示该位置。

查询结果存放在　answer 列表里。

算法：

```
While p1 != null AND p2 != null
  If p1->docId == p2->docId //  对两（剩余）列表的首元素进行比较
    While q1 != null AND q2 != null
      If q1->pos –q2->pos<=k OR q2->pos –q1->pos <= k
        insert(answer, p1);
        break; // 跳出这个循环，找到一个　　k 临近即可
```

```
        ElseIf q1->pos –q2->pos> k      //q2 不可能被匹配上，忽略它
            q2= q2->next;// 生成新的剩余列表
        Else If q2->pos –q1->pos > k    //q1 不可能被匹配上，忽略它
            q1=q1->next;// 生成新的剩余列表
    End If
    End While
    p1=p1->next; // 构造新的剩余列表，迭代执行
    p2=p2->next;
Else if p1->docId < p2->docId
    p1=p1->next ；  //p1->docId 不可能有匹配；构造新的剩余列表
Else
    p2=p2->next ；  //p2->docId 不可能有匹配；构造新的剩余列表
End
```

第六章　文档评分、词项权重计算及向量空间模型

习题 **6-2**

上面的例 6-1 中，如果 g1 = 0.2, g2 = 0.31 及 g3 = 0.49，那么对于一个文档来说所有可能的不同得分有多少？

得分 1:　0
得分 2：g1=0.2
得分 3：g2=0.31
得分 4：g3=0.49
得分 5：g1+g2=0.51
得分 6：g1+g3=0.69
得分 7：g2+g3=0.8
得分 8：g1+g2+g3=1.0

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（这里改为 df 值的计算就假设用 Doc1, Doc2 Doc3 的这个文集 ）。

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

图 6-9　习题 6-10 中所使用的 tf 值

解答：

$w_{t,d}=max(1+log_{10}(1+tf),0)$			
	Doc1	Doc2	Doc3
Car	2.4314	1.6021	2.3802
Auto	1.4771	2.5185	0
insurance	0	2.5185	2.4624
Best	2.1461	0	2.2304

	df_t	idf_t
car	3	0
auto	2	0.1761
insurance	2	0.1761
best	2	0.1761

这里 N=3。

$tf\text{-}idf_{t,d}= w_{t,d}*idf_t$			
	Doc1	Doc2	Doc3
car	0	0	0
auto	0.2601	0.4435	0
insurance	0	0.4435	0.4336
best	0.3779	0	0.3928

例 **6-4**

假设文档集中的文档数目 N=1000000，词表为 {auto, best, car, insurance} ，这四个词的 df 值分别为 5000, 50000, 10000, 1000。
设某文档 d 的 raw tf 向量为 [1,0,1,2]，对查询 q=”bestcar insurance ,” 问该文档 -查询的相似度打分 score(q,d)是？

解答：

	df_t	idf_t
auto	5000	2.3010
best	50000	1.3010
car	10000	2.0000
insurance	1000	3.0000

这里 N=1000000。

文档 d 的 tf-idf 向量：

	raw $tf_{t,d}$	$w_{t,d}=max(1+log_{10}(1+tf),0)$	$tf\text{-}idf_{t,d}= w_{t,d}*idf_t$	v(d)= 归一化 $tf\text{-}idf_{t,d}$
auto	1	1.0000	2.3010	0.4646
best	0	0	0	0
car	1	1.0000	2.0000	0.4038

insurance	2	1.3010	3.9031	0.7881

查询 q 的 tf-idf 向量（ $w_{t,d}$ =1）

	raw tf $_{t,q}$	$w_{t,q}$=max(1+log$_{10}$(1+tf),0)	tf-idf $_{t,q}$= $w_{t,q}$*idf $_t$	v(q)= 归一化 tf-idf $_{t,d}$
auto	0	0	0	0
best	1	1	1.3010	0.3394
car	1	1	2.0000	0.5218
insurance	1	1	3.0000	0.7827

score(q,d) = v(q) *v(d)=0.8275

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-precision 值，并与　a 中按照　MAP 进行排序的结果进行对比。

解答：

a. 按 MAP 的定义，这里　|Q|=1　，　m=4。在查询结果中遇到每个相关文档对前面的所有文档计算一个　Precision，　MAP 将这些　Precision 值求平均。

   MAP(系统 1)= (1/4)*(1+2/3+3/9+4/10) = 0.6

   MAP(系统 2)= (1/4)*(1/2+2/5+3/6+4/7)=0.49

   系统 1 的 MAP 值大。

b. 相关的查询结果排名越靠前，则　　MAP 越大。

c. 按　R-precision　的定义，假设总共有　|Rel| 篇相关文档，在查询结果中取前　　|Rel| 个文档，计算其　precision。

   R-precision(系统 1)=2/4=1/2

   R-precision(系统 1)=1/4

   系统 1 的 R-precision 值大。与　MAP 给出系统打分排序的结果一致。

习题 **8-10**

下表中是两个判定人员基于某个信息需求对　　12 个文档进行相关性判定的结果（　　0=不相关，　1=相关）。假定我们开发了一个　　IR 系统，针对该信息需求返回了文档　　{4, 5, 6, 7, 8}。

docID	判断 1	判断 2
1	0	0
2	0	0

3	1	1
4	1	1
5	1	0
6	1	0
7	1	0
8	1	0
9	0	1
10	0	1
11	0	1
12	0	1

a. 计算两个判断之间的 kappa 统计量；

b. 当两个判断均认为是相关文档时才认为该文档相关，此时计算上述系统的正确率、召回率及 F_1 值；

c. 只要有一个判断认为是相关文档则认为该文档相关，此时计算上述系统的正确率、召回率及 F_1 值。

解答：

a. 计算 kappa 统计量：

P(A) 就是实际观察到的一致意见的概率，总共 12 篇文档，其中 2 篇两人一致选 Yes，2 篇两人一致选 No。因此，P(A)=(2+2)/12=1/3 。

P(E)是随机情况下的一致意见的概率。假设每个人对每个文档的 Yes（或 No）打分的概率 p_y （或 p_n ）是独立同分布的 (i. i. d.)，则 $P(E)= p_y*p_y + p_n*p_n$。其中， p_y 是 2*12 次打分中为 Yes的比例， $p_y=12/24=1/2$ ；p_n 是 2*12 次打分中为 No 的比例， $p_n=12/24=1/2$ 。代入 P(E)，得： P(E)=(1/2)^2+(1/2)^2=1/2 。

Kappa=(P(A)-P(E))/(1-P(E))=(1/3-1/2)/(1-1/2)=-1/3<0.67 ，这是一个负数，说明实际的一致性结果还不如随机产生的一致性结果，因此可以判定两人给出的相关性打分不一致。

b. 文档集中共有 12 篇文档，其中 2 文档相关（ {3,4} ），其它 10 篇都不相关。查询结果为 {4, 5, 6, 7, 8}，其中只有 1 篇文档相关（ {4}）。

该查询的

Precision, P=1/5 ；

Recall, R=1/2 ；

F_1=2P*R/(P+R)=0.28。

c. 文档集中共有 12 篇文档，其中 10 文档相关，其它 2 篇都不相关（ {1,2} ）。查询结果为 {4, 5, 6, 7, 8}，全部都相关。

该查询的

Precision, P=1 ；

Recall, R=5/12 ；

F_1=2P*R/(P+R)=0.67。

注：因 Kappa 统计量认为两人打分不一致， 所以修正方法 b 比较合理， 而 c 非常不合理。

第十三章　文本分类与朴素贝叶斯方法

习题　**13-3**

位置独立性假设的基本原则是，　　词项在文档的位置　k 上出现这个事实并没有什么有用的信息。请给出这个假设的反例。提示：可以考虑那些套用固定文档结构的文档。

解答：如果一个词出现在不同域中，它的重要性不同。比如出现在标题中的词一般很重要。

习题　**13-9**

基于表　13-10 中的数据，进行如下计算：

(i)　估计多项式　NB 分类器的参数；

(ii)　将 (i)中的分类器应用到测试集；

表13-10　用于参数估计的数据

	文档ID	文档中的词	属于c=China类?
训练集	1	Taipei Taiwan	yes
	2	Macao Taiwan Shanghai	yes
	3	Japan Sapporo	no
	4	Sapporo Osaka Taiwan	no
测试集	5	Taiwan Taiwan Sapporo	?

P(China)=2/4=1/2; P(非 China)=2/4=1/2.

词典中有　7 个词 Japan, Macao, Osaka, Sapporo, Shanghai, Taipei, Taiwan.

测试集中，　China 类共有　5 个词；非　China 类共有　5 个词。

P(Taiwan|China 类)=(2 + 1)/(5 + 7)= 1/4　（加一平滑，下同）

P(Taiwan|非 China 类) =(1 + 1)/(5 + 7)= 1/6

P(Sapporo|China 类)= (0 + 1)/(5 + 7)= 1/12

P(Sapporo|非 China 类)= (2 + 1)/(5 + 7)= 1/4

按单字词语言模型，

P(China 类 $|d_5$) $\propto$ P(China 类)* P(Taiwan|China 类)^2* P(Sapporo|China 类)=1/2*(1/4)^2*1/12=1/384.

P(非 China 类 $|d_5$) $\propto$ P(非 China 类)* P(Taiwan|非 China 类)^2* P(Sapporo|非 China 类)=1/2*(1/6)^2*1/4=1/288.

由于 P(非 China 类$|d_5$)> P(China类$|d_5$)，d_5属于非 China 类。

第十六章　扁平聚类

习题　**16-3**　对于图　16-4，同一类中的每个点　d 都用两个同样的　d 的副本来替换。　(i)　那么，对于新的包含　34 个点的集合进行聚类，　会比图　16-4 中　17 个点的聚类更容易、　一样难还是更难？ (ii)　计算对　34　个点聚类的纯度、　RI。在点数增加一倍之后，哪些指标增大？哪些指标保持不变？　(iii)　在得到 (i)中的判断和　(ii)中的指标之后，　哪些指标更适合于上述两种聚类结果的质量比较？

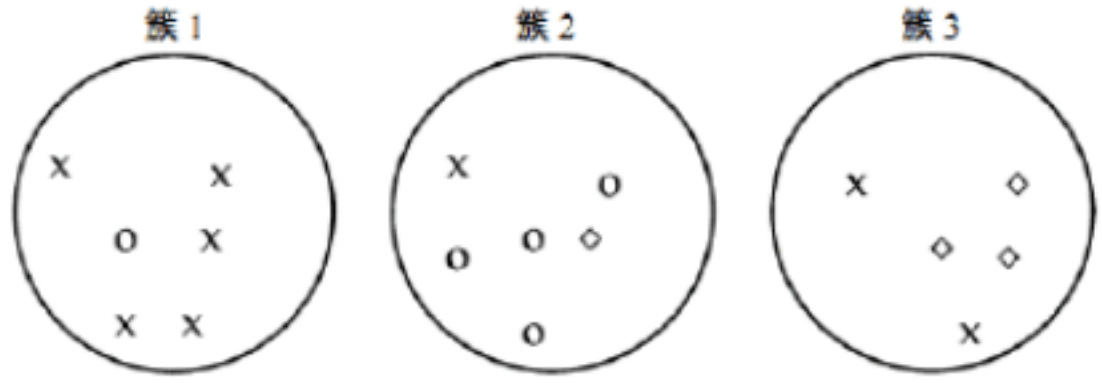

图 16-4　作为度量簇质量的外部准则的纯度示例。每个簇所对应的主类及属于其的文档数目分别是：X, 5（簇 1）；O, 4（簇 2）；及◇, 3（簇 3）。纯度为(1/17) × (5 + 4 + 3) ≈ 0.71

解答：

（i）

我认为更难，因为　34 个点比 17 点的计算量增大了。

（ii）

节点复制为原先的一倍后，

簇 1：10 个 x 类文档，　2 个 o 类文档；

簇 2：2 个 x 类文档，　8 个 o 类文档，　2 个◇类文档；

簇 3：4 个 x 类文档，　6 个 ◇类文档。

共有 N=34 篇文档。

计算纯度 =(1/34)*(10+8+6) ≈0.71；

计算 RI：

$TP = \binom{10}{2} + \binom{2}{2} + \binom{2}{2} + \binom{8}{2} + \binom{2}{2} + \binom{4}{2} + \binom{6}{2} = 97$，将一对同类的文档分到相同聚类中的对数。

TN = 10 ? 8 + 10 ? 2 + 2 ? 2 + 2 ? 2 + 10 ? 6 + 2 ? 4 + 2 ? 6 + 2 ? 6 + 8 ? 4 + 8 ? 6 + 2 ? 4 = 288，将一对不同类的文档分到不同聚类中的对数。

$RI = (TP + TN)/\binom{34}{2} = (97 + 288)/561 \approx 0.69.$

（iii）

对比 N=17 时，纯度为　0.71，RI 为 0.68。我们得出节点复制为原先的一倍后，指标几乎不变。

习题 **16-4**　在 K-均值算法中，为什么对同一概念　car 使用不同词项来表示的文档最后可能会被归入同一簇中？

解答：

考虑两篇文档，　一篇含有　car 和其它词，　一篇含有　automobile 和其它词。　虽然第 2 篇文档不含　automobile，但这两篇文档可能含有大量的公共词，它们的文档向量可能是相似的，聚类算法将把它们分配到同一簇中。

习题 **16-5**　K-均值算法的两个停止条件为：　(i) 文档的分配不再改变；　(ii) 簇质心不再改变。请问这两个条件是否等价？

解答：

连续两次迭代，文档到分配簇的情况不再改变，说明簇质心的计算也不再改变。

连续两次迭代，簇质心不再改变，按照最近距离原则，文档到分配簇的情况也不再改变。

因此，条件 (i)和 (ii) 是等价的。