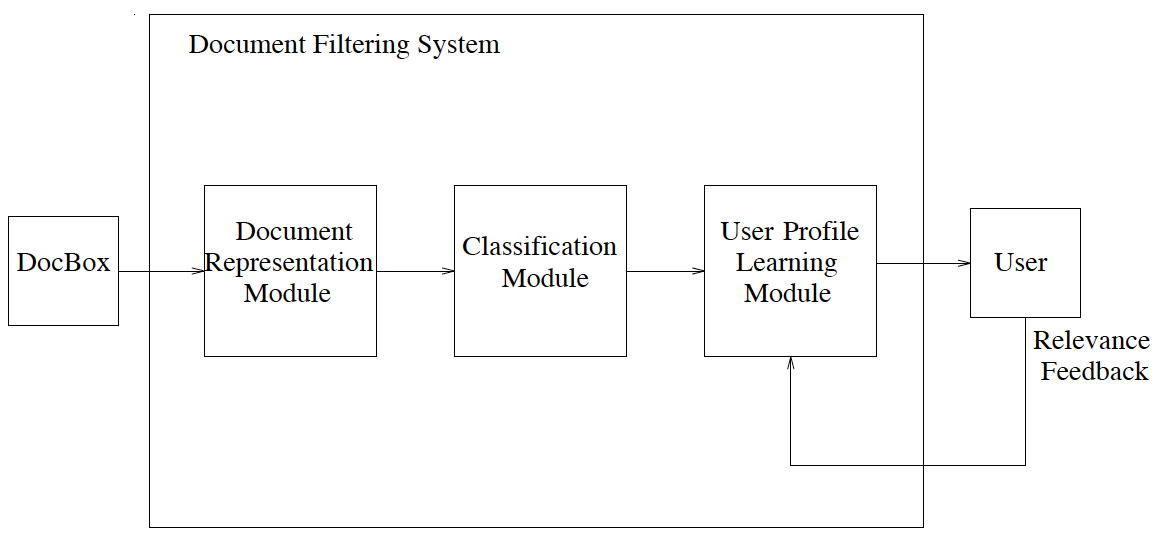
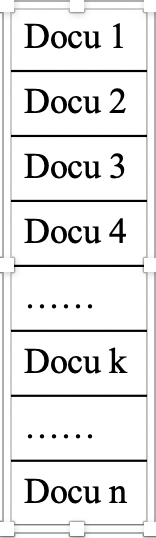
流程：



系统一共分为四个模块：

1. 第一个模块（document acquisition module DAM）：

从数据库中下载6000篇文献，离线保存在DAM模块中。n = 6000

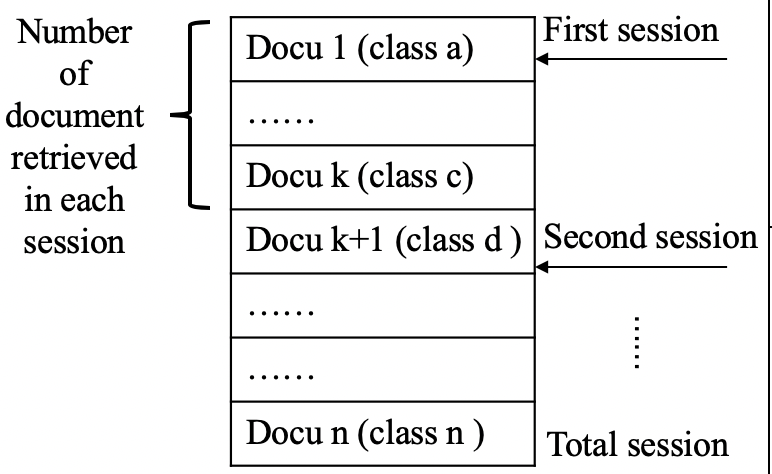


从这6000篇文献中，随机抽取2000篇作为代表性文献。

词库选择的是：，该词库中一共有M个词。文档中的词包含在词库中，因此定义一个M维向量（M等于词典大小），若文档中的某个词在词典中出现，就给这个值计算tfidf，若没有出现，则在相应位置赋值为0.

1. 第二个模块（document representation module）：

每次从DAM中抽取20篇文献，并将每篇文献采用vector-space-model（向量空间法）表示。如下图，每个session=20，6000篇文章的话，总共有300个session。



其中，idf的计算是以2000篇代表文献为基础的。Tf按照每次抽取的每篇文章计算。

1. 第三个模块（document classification module）：

首先，以2000篇文档为基础，形成n个类别。使用启发式、无监督的聚类算法：Maximin-distance。

接着，对于每次抽取的20篇文档进行分类，使用cosine similarity measure方法。每篇文档属于上述类别中的某一类。

1. 第四个模块（user profile learning模块）：

首先，设置两个向量 d和p。大小都等于类别数n。

初始值设置为：

d =（ 类别1 : 0；类别2 : 0；类别3 : 0； ……; 类别n：0）；即所有类别初始值为0；

p =（ 类别1 : ；类别2 : ；类别3 : ； ……; 类别n：）；所有类别初始值为1/n；

设计用户界面：

1. 首页：登陆界面，有账号的登陆账号，没有的建立账号和密码，或者也可以选择不建立，直接跳过。界面类似于下图：



图1

1. 每个新的用户登陆进来之后出现一下界面：

兴趣值：

1: 类别1；

2: 类别2；

3: 类别3；

……

n: 类别n；

注：兴趣值在0-1之间，越大表示用户对这类话题的兴趣值越高。

Submit

Skip

图2

方框中用户可以填写对于某个类别的兴趣值，0-1之间。如果用户选择skip，则d和p的值依旧等于初始值；

如果用户用户填写了兴趣值，则开始更新d和p。假设一共就只有五个类别（用户填写的类别1的兴趣值为0.1， 类别2的兴趣值为0.2， 类别3的兴趣值为0.6），其余依旧为0.

则 更新d = （类别1 : 0.1；类别2 : 0.2；类别3 : 0.6； ……; 类别n：0）

更新p: 其中，k+1指的是下一时刻，i = （1，2，…，n）。其中，E为单位向量，d中类别最大值设为为1，其余为0。即在本次假设中，类别3的值最大，为0.6. 则E=（0，0，1，0，0）. 是0-1之间的值，可以修改，这边假设为0.5. 具体为：

假设一共有5个类别

p初始 = （1/5； 1/5； 1/5； 1/5；1/5）

p下一时刻 = （1/5，1/5，1/5，1/5，1/5）+ 0.5 【（0，0，1，0，0）- （1/5，1/5，1/5，1/5，1/5）】

= （1/10， 1/10， 6/10， 1/10，1/10）= （0.1，0.1，0.6，0.1，0.1）

1. 对抽取的20个文档，根据其类别以及d和p的值进行排序。比如抽取出来的20个文档中，前10个文档属于类别1，11-15个文档属于类别3，16-20个文档属于类别4。那么，

首先，利用p产生一个排在top的类别。具体做法为：产生一个从0-1之间的伪随机数（pseudo-random number PRN），如果PRN的值在【0-0.1）之间，那么类别1将会被排在最前面，如果PRN的值在【0.1-0.2）之间，那么类别2将会被排在最前面；如果PRN的值在【0.2-0.8）之前，那么类别3将会被排在最前面；如果PRN的值在【0.8-0.9）之间，类别4排在最面前，【0.9-1.0）之间，第五个类别在最前面。

假设PRN产生的随机数是0.6，那么第三个类别排在最前面，即11-15个文档排在最前面。

接着，查询d的值，对于剩下的类别，按照d值的大小排序，即接着是类别1，最后是排类别4. 注意：如果用户直接点击skip，那么d和p的值依旧等于初始值，p=（0.2，0.2，0.2，0.2，0.2），还是产生一个PRN来决定属于哪一个类别的文档排在最前面，剩下的文档就根据d来排序，若是d想同，则按照文档流的顺序排列即可。

简化的界面如下图所示：

Click on the subject lines that you wish to read

1. 文档11
2. 文档12
3. …….
4. 文档15
5. 文档1
6. 文档2
7. ……
8. 文档10
9. 文档16
10. 文档17
11. ……
12. 文档20

* X

Profile

Click to exit

图3

若左上角的Profile按钮被点击之后，会出现当前d的值，即在本次案例中：d = （类别1 = 0.1；类别2= 0.2；类别3 = 0.6； 类别4=0 ; 类别5=0）如下图所示（类别---兴趣值），

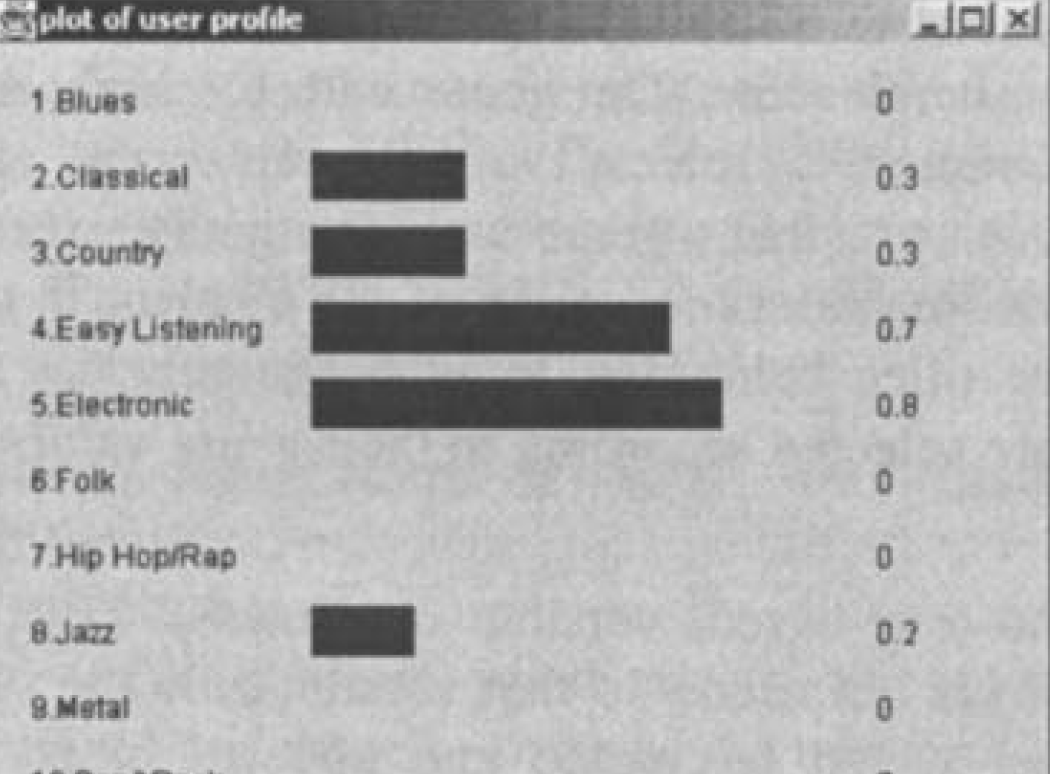


图4

1. 用户如果点击上述某一个文档，则出现如下界面，即显示该文档的具体内容，并且用户可以提交界面下方的0.1-1.0之间的值，表示对这篇文档的相关性的判断，点击submit提交，阅读完文档之后，点击back按钮之后，回到3）中的页面，又可以选择查看另一篇文档。：

文档内容。。。。。。

。

。

。

。

。

。

Use the scale to indicate relevance, the hit the <back> button

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 | 1.0 |

back

Submit

图5

1. 根据用户提交的反馈值，更新d，进而更新p。更新d的算法如下，更新p的算法如2）中所示。

d = 其中，n是总数。d是累积值，比如已经进行到第10个session了，那么就是，即第一个session结束后d的值 + 第二个session结束后d的值 + 。。。。。。 +第9个session结束后d的值 + 本次session得到的反馈值。

举例说明：d的当前值为：（类别1 : 0.1；类别2 : 0.2；类别3 : 0.6；类别4：0；类别5: 0）.

假如用户给文档11的反馈值是0.5，给文档14的反馈值是0.7，给文档16的反馈值是0.1.

因为文档11和14都属于类别3，在该session中，用户给类别3的平均反馈值 = = 0.6； 文档16属于类别4，则d中，类别3的值更新为：

d3 = =，

类别4的值更新为： d4 = =

此时的d的值更新为：（类别1： ； 类别2 ： ； 类别3： ； 类别4: ； 类别5: ；）

此时d中，依旧是类别3的值最大，因此，新的p为：

= （0.1，0.1，0.6，0.1，0.1）+ 0.5 【（0，0，1，0，0）- （0.1，0.1，0.6，0.1，0.1）】

= (0.05, 0.05, 0.8, 0.05, 0.05)

1. 根据上述算法，每进行完一个session，更新一次d和p。
2. 当用户在图3中没有想要浏览的文档之后，点击click to exit的按钮，会离开当前界面，表示当前的session结束，然后进入如下图5界面：该界面与图3界面文档的排列顺序一样，主要目的是获得评估系统（不对d或者p的值产生任何影响）。在这个界面中，用户主要是勾选该页中与其相关的文档。

1文档11

2文档12

3…….

4文档15

5文档1

6文档2

7……

8文档10

9文档16

10文档17

11……

12文档20

图6

勾选完成之后，计算这个session的准确率，公式如下：

Precision = 1 -

其中，N是文档流的总数，即在假设中每次session取的20篇文档。REL表示相关文档的总数，也就是图6中勾选的相关文档数；RANKi 是与用户相关的文档i在图3中的排序。

将上述结果保存在excel中（或者txt），方便查看每次session结束后的准确率的值。

计算完之后，进入第二个session。即从文档库中重新抽取20篇文档，根据d和p的值，再次对新的文档进行排序，然后如上图3一样，呈现给用户。重复上述过程。

1. 只有在用户点击的是图2还是图3中的右上角的 ‘X’，才表示用户想要离开系统。点击之后都将出现如下界面：

1: 类别1；

2: 类别2；

3: 类别3；

……

n: 类别n；

图7

该界面要求用户在离开系统之前填写目前关于各个类别的兴趣值，并将用户每次离开系统前填写的兴趣值与当前系统获得的d的值放在一起，用于比较两者之间的差异，即系统当前获得的值与用户目前的真实值之间的差异。比如，放在一张excel表中：

|  |  |  |
| --- | --- | --- |
| 第一次离开系统时： | 用户填写的兴趣值 | 系统获得的兴趣值（也就是图3中profile显示出来的值） |
| 类别1 |  |  |
| 类别2 |  |  |
| 。。。。。。 |  |  |
| 类别n |  |  |
| 第二次离开系统时 |  |  |
| 。。。。。。 |  |  |

当用户填写完之后，退出系统。

注意：用户每次点击图2和3的‘X’时，都将表示离开系统，离开之前都要完成图6.并在文档中保存值。

8） 比如用户离开界面时，已经完成了30个session，那么下一次进入系统时，从第31个session开始，并且d和p的值为第30个session结束时的值，而不是初始值。从第31个session开始，在文档库中抽取20篇文献，重复document representation，classification以及profile learning模块。向用户继续推荐文档，获得用户的反馈，更新d和p，直到所有session都完成。