第七章：

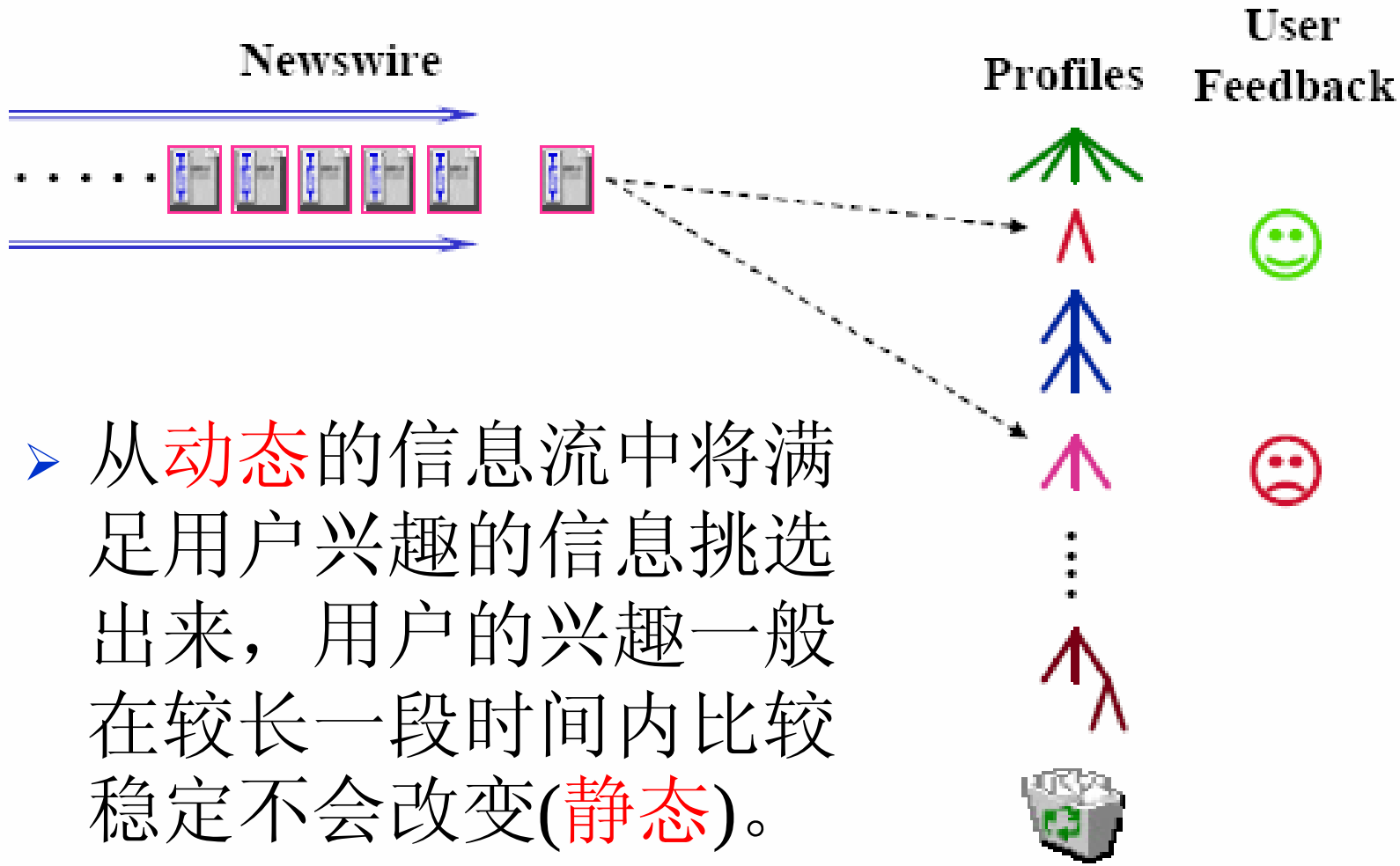
文本过滤技术

杨建武

北京大学计算机科学技术研究所

Email:yangjianwu@icst.pku.edu.cn

信息过滤的定义



- 从动态的信息流中将满足用户兴趣的信息挑选出来，用户的兴趣一般在较长一段时间内比较稳定不会改变(静态)。

信息过滤系统的特点

- 新信息的产生速度很快，相对来说，人的兴趣变化比较缓慢，可以看成相对静态的和稳定的。
- 信息过滤主要借用信息检索和用户建模(User modeling)两个领域的技术。
- 用户的需求或者兴趣通常采用User Profile建模来表示。
- 新信息到来的时候，根据用户的User Profile，有选择地挑出信息给用户。

信息过滤与信息检索

- 信息过滤(IF)信息检索(IR)
- IF是可以看成广义IR的一部分，即和Adhoc Retrieval相对的一种任务模式。
- IR通常采用Pull模式，而IF通常采用Push模式。
- 和Adhoc Retrieval相比：
  - IF信息源动态，用户需求(采用User Profile来表示)相对静态；
  - IR信息源相对静态，用户需求(采用Query来表示)动态变化
  - IR可以认为面向一次性的查询而使用，而IF是面向用户的长期需求的重复使用
  - IF一般要关注用户建模，涉及用户隐私问题，而IR一般不需要。

信息过滤与信息分类

- IF vs. IC (Info. Classification)
- IF可以采用IC中的分类算法。
- 某些场合下人们所称的“信息过滤”实际就是一个IC问题。如不经过用户Profile调整的垃圾邮件过滤。
- IC中的Category通常不会变化，相对而言，IF的User Profile会动态调整。

信息过滤与信息提取

- 信息提取(Information Extraction, IE)是从无格式数据源中抽取相关字段的过程。比如抽取恐怖事件的时间、地点、人物等字段。
- IE中不太关注相关性，而只关注相关的字段。IF中要关注相关性。

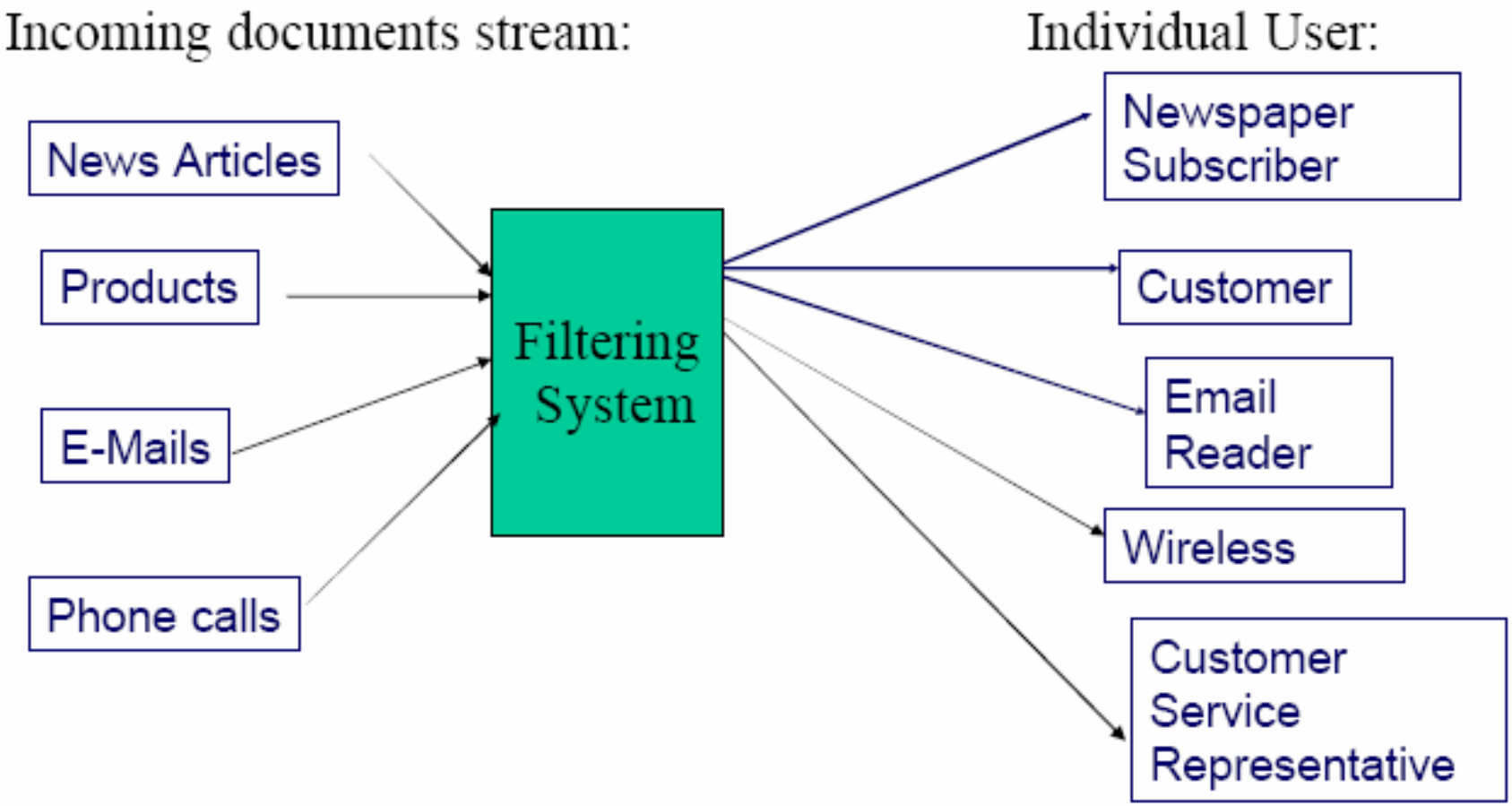
Other info. seeking processes

Process	*Information Need*	*Information Source*
Data Bases	Dynamic & Specific	Stable & Structured
Information Retrieval	Dynamic & Specific	Stable & Unstructured
Information Filtering	Stable & Specific	Dynamic & Unstructured
Alerting	Stable & Specific	Dynamic
Information Extraction	Specific	Unstructured
Browsing	Broad	Unspecific

信息过滤的一些应用

- 克服重复查询
  - 网络信息是动态变化的, 用户时常关心这种变化
  - 而在搜索引擎中, 用户只能不断地在网络上查询同样的内容,以获得变化的信息, 这花费了用户大量的时间
- 提供个性化信息服务
  - 对不同的用户采取不同的服务策略, 提供不同的服务内容。
  - 实现“主动服务”，“信息找人”
- 实现有害信息的过滤
  - 反动言论，保护国家安全
  - 谣言，保护社会稳定
  - 色情内容，保护青少年身心健康

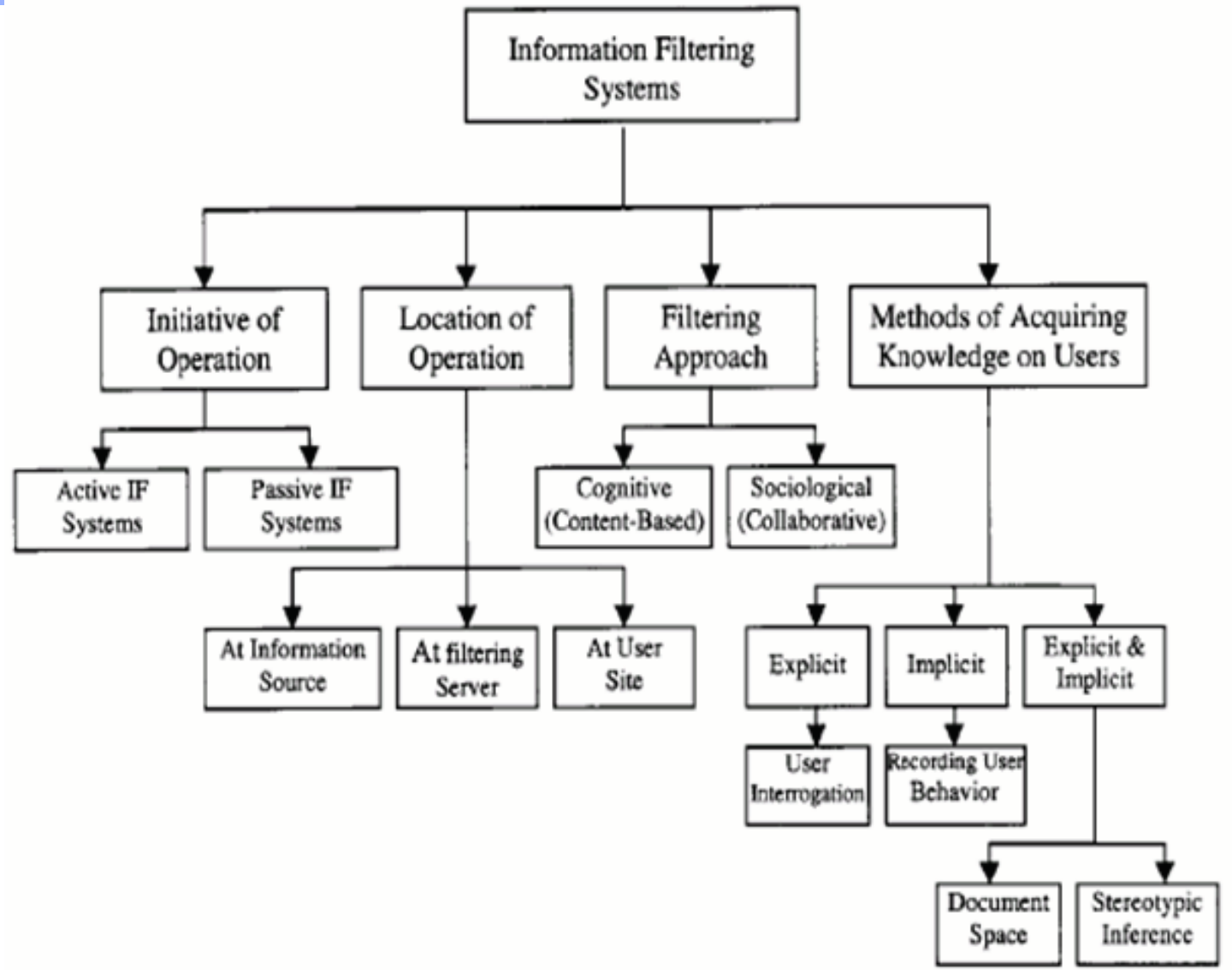
信息过滤的一些应用


Incoming documents stream:
Individual User:
News Articles
Products
E-Mails
Phone calls
Filtering System
Newspaper Subscriber
Customer
Email Reader
Wireless
Customer Service Representative

信息过滤的一些应用

- 搜索引擎检索结果的过滤：Google
- 个人的邮件过滤
- 新闻订阅和过滤
- 浏览器过滤
- 面向儿童的过滤系统
- 面向客户的过滤系统和推荐系统

IF分类示意图


Information Filtering Systems
Initiative of Operation
Location of Operation
Filtering Approach
Methods of Acquiring Knowledge on Users
Active IF Systems
Passive IF Systems
Cognitive (Content-Based)
Sociological (Collaborative)
At Information Source
At filtering Server
At User Site
Explicit
Implicit
Explicit & Implicit
User Interrogation
Recording User Behavior
Document Space
Stereotypic Inference

按Initiative of operation分

- 主动(Active)的IF系统
  - 主动搜集信息，并将相关信息发送给用户
  - 通常采用Push操作
  - 会造成信息过载问题，所以该系统要尽力建立精确的User Profile。
  - 代表系统BackWeb
- 被动(Passive)的IF系统
  - 不负责为用户搜集信息
  - 通常用于邮件和新闻组信息过滤
  - 代表系统GHOSTS

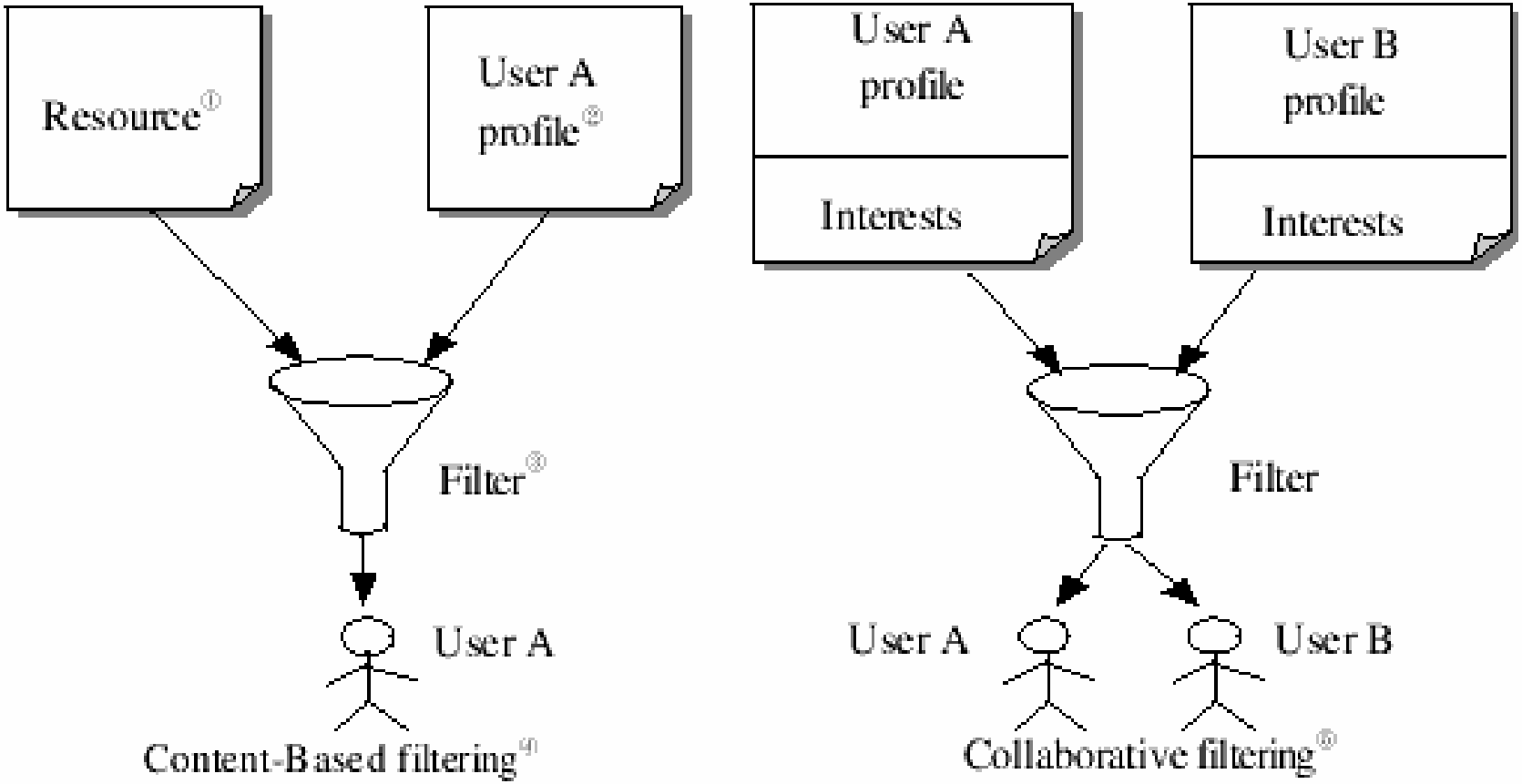
按Location of operation分

- 在信息源端过滤
  - 将用户的Profile发送给信息提供者，后者将和用户Profile匹配的信息回送给用户
  - 用户通常需要付费，
- 在过滤服务器端过滤
  - 信息提供者将信息发送给过滤服务器
  - 过滤服务器根据用户的Profile将匹配信息发给用户
- 在用户端过滤
  - 是一个局部过滤系统
  - 如outlook的过滤功能。

从过滤方法分

- 基于感知的过滤(Cognitive filtering)
  - 也称为基于内容的过滤(Content-based filtering)
  - 将文档内容和用户的Profile进行相似度计算
- 基于社会的过滤(Sociological filtering)
  - 也称为协同过滤(Collaborative filtering
  - 对某个用户的Profile进行匹配时，通过用户之间的相似度来计算Profile和文档的匹配程度
  - 基于社会过滤的系统常常称为推荐系统(Recommendation systems)
  - 社会过滤常常使用用户建模(User modeling)及用户聚类(User clustering)等技术。
  - 社会过滤一般不单独使用，常常和基于内容的过滤配合使用。

基于内容的过滤与协同过滤



①资源,②描述文件,③过滤器,④基于内容的过滤,⑤协作过滤.

获取用户信息

- 显式获取用户信息
  - 用户直接填表
  - 用关键词表达用户过滤需求
  - 用文档集表达用户过滤需求
- 隐式获取用户信息
  - 无需用户直接参与，通过观察用户的动作行为判断用户需求
  - 用户阅读文档的时间可以作为衡量该文档相关度的一个指标。
  - 其他的一些用户行为——诸如用户是否保存、删除或是打印某篇文档也可以作为度量文档相关度的一个指标。

评估指标

评估指标

- 正确率和召回率(Precision & Recall)
- 基于统计的评价指标
  - 相关系数(Correlation)：用户评估的结果排序和系统评估的结果排序的序相关系数
- 基于集合的评价指标
  - Utility=(A* R^+)+(B* N^+)+(C* R^-)+(D*N^-)

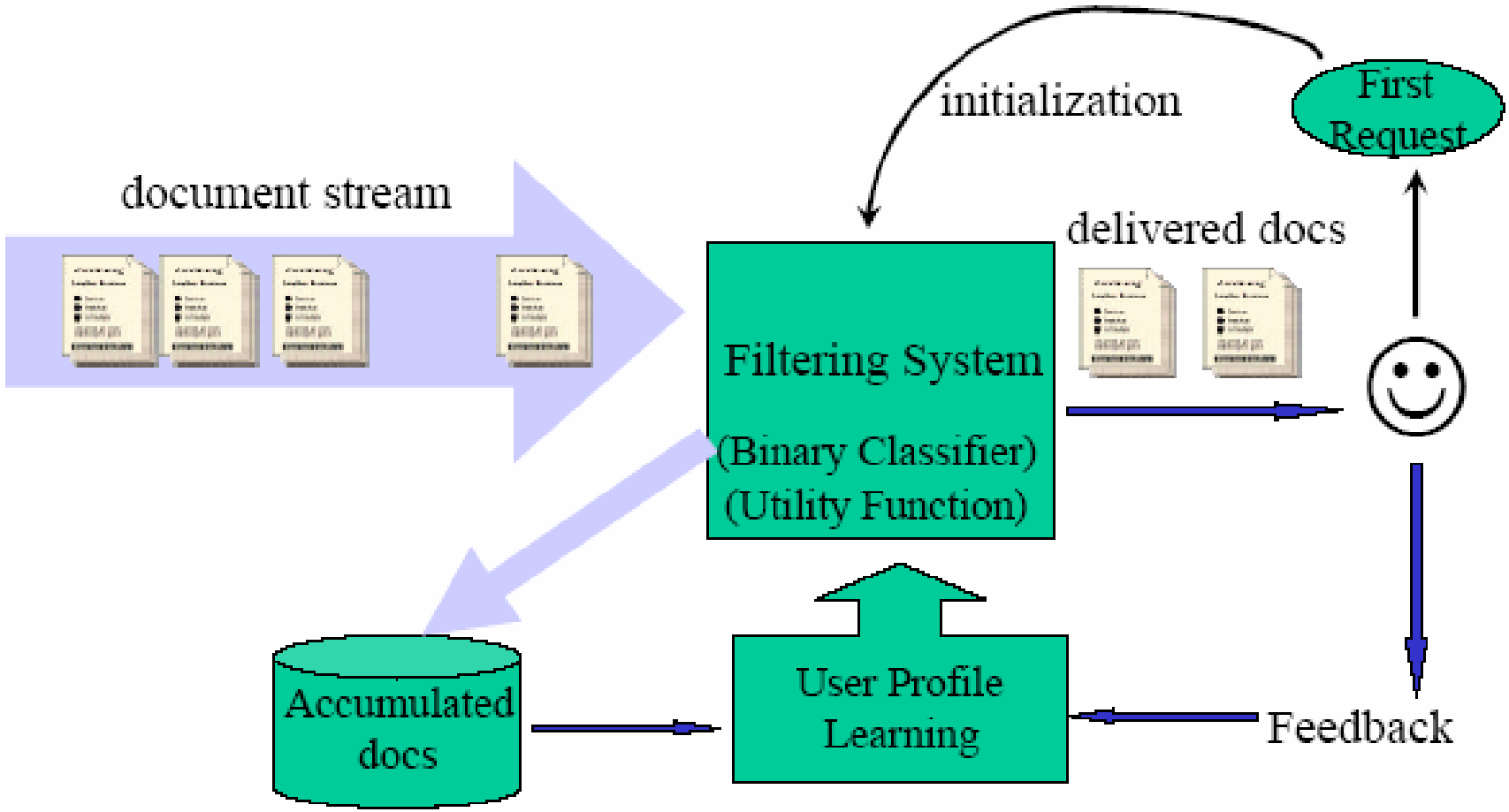
    R^+ N^+ R^- N^-分别表示选出来的结果中真正相关文档的个数、不相关文档的个数、未选出来结果中相关文档的个数及不相关文档的个数，A、B、C、D是加权系数。
  - ASP(average set precision)=P*R, 当P or R=0, ASP 不可用

评估指标

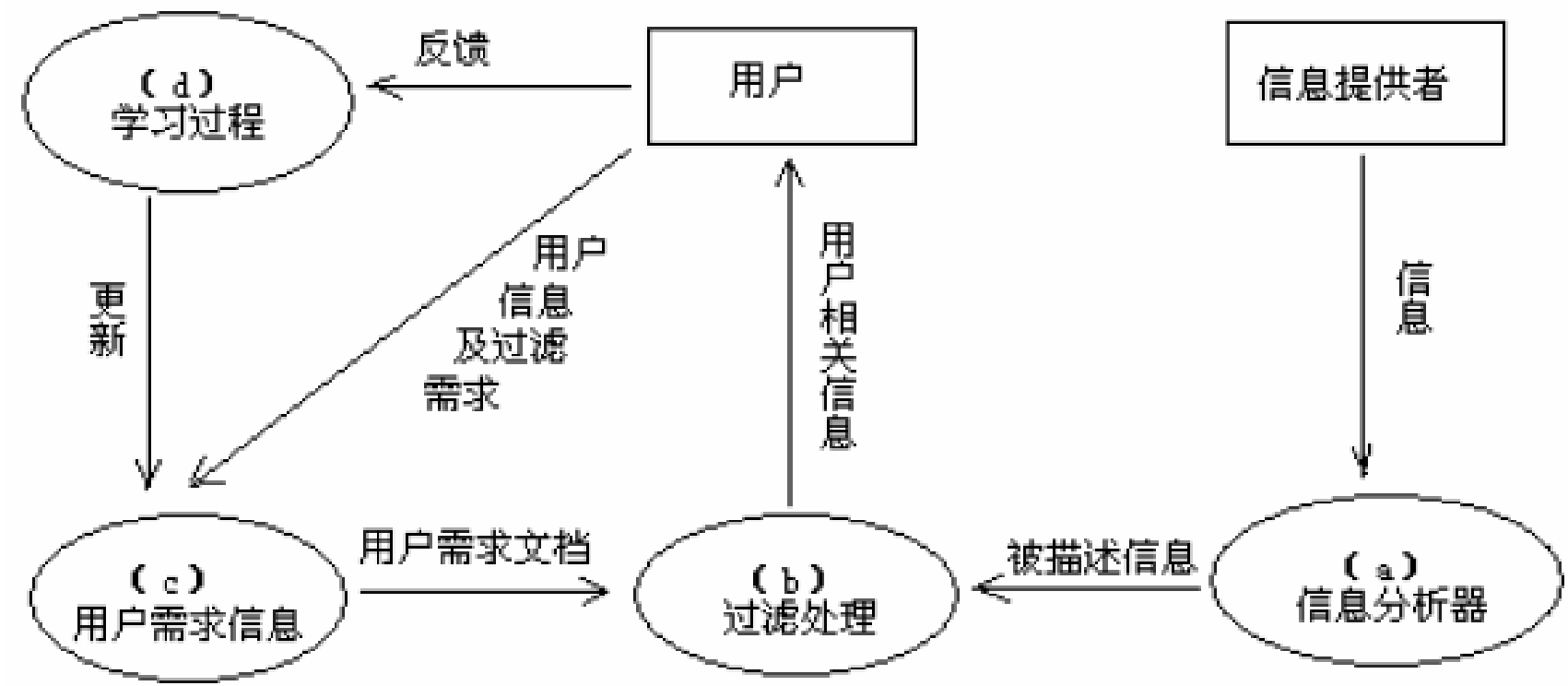
- 面向用户(User-oriented)的指标
  - Coverage Ratio=$|R_k|/|U|=|A \cap U|/|U|$,
    R_k是用户已知的相关文档集合
  - Novelty=$|R_u|/(|R_u|+|R_k|)$,
    R_u是系统找出的用户未知的相关文档集合

主要方法

A Typical Adaptive Filtering System



过滤系统


（d）
学习过程
反馈
用户
信息提供者
更新
用户
信息
及过滤
需求
用户
相关
信息
信息
（c）
用户需求信息
用户需求文档
（b）
过滤处理
被描述信息
（a）
信息分析器

数据信息分析部分

- 靠近信息提供者
- 从信息提供者处获得或者搜集数据
- 文档分析或表示(例如：Boolean Model,VSM, etc)
- 把这种表示传给过滤部分

用户模型部分

- 收集用户信息(显式或者隐式的)
- 构造用户profiles或者其它的用户模型(rules, VSM, documents center)
- 把用户模型也传给过滤部分
- 用户模型必须适应文档表示

过滤部分

- 信息过滤系统(IF system) 的核心
- 将用户的profiles和数据项的表示相匹配
- 最终决策可能是二元的或者一个概率值(可以排序)
- 相关信息被送到学习部分(feedback)

学习部分

- 为了提高过滤系统的性能
- 侦测用户兴趣的变化
- 更新用户模型

基于查询的方法

- The Rocchio algorithm
- Represent each delivered document by a vector
  - The usual approach: Lexical processing, tf.idf weights, etc
- Represent the query by a vector
  - The usual approach
- The modified query is a weighted average of the original query, and relevant and non-relevant document vectors (反馈学习)
  - R is the set of relevant documents
  - R is the set of non-relevant documents

$$Q' = Q + \alpha \frac{1}{|R|} \sum_{D_j \varepsilon R} D_j - \beta \frac{1}{|NR|} \sum_{D_j \varepsilon NR} D_j$$

基于查询的方法--Example

Original Query:

(5, 0, 3, 0, 1)

Document D1, Relevant:

(2, 1, 2, 0, 0)

Document D2, Non-relevant:

(1, 0, 0, 0, 2)

$\alpha = 0.50, \beta = 0.25$

Relevance Feedback Formula:

$$Q' = Q + \alpha \frac{1}{|R|} \sum_{D_j \in R} D_j - \beta \frac{1}{|NR|} \sum_{D_j \in NR} D_j$$

Q' = Q + 0.5 D1 - 0.25 D2

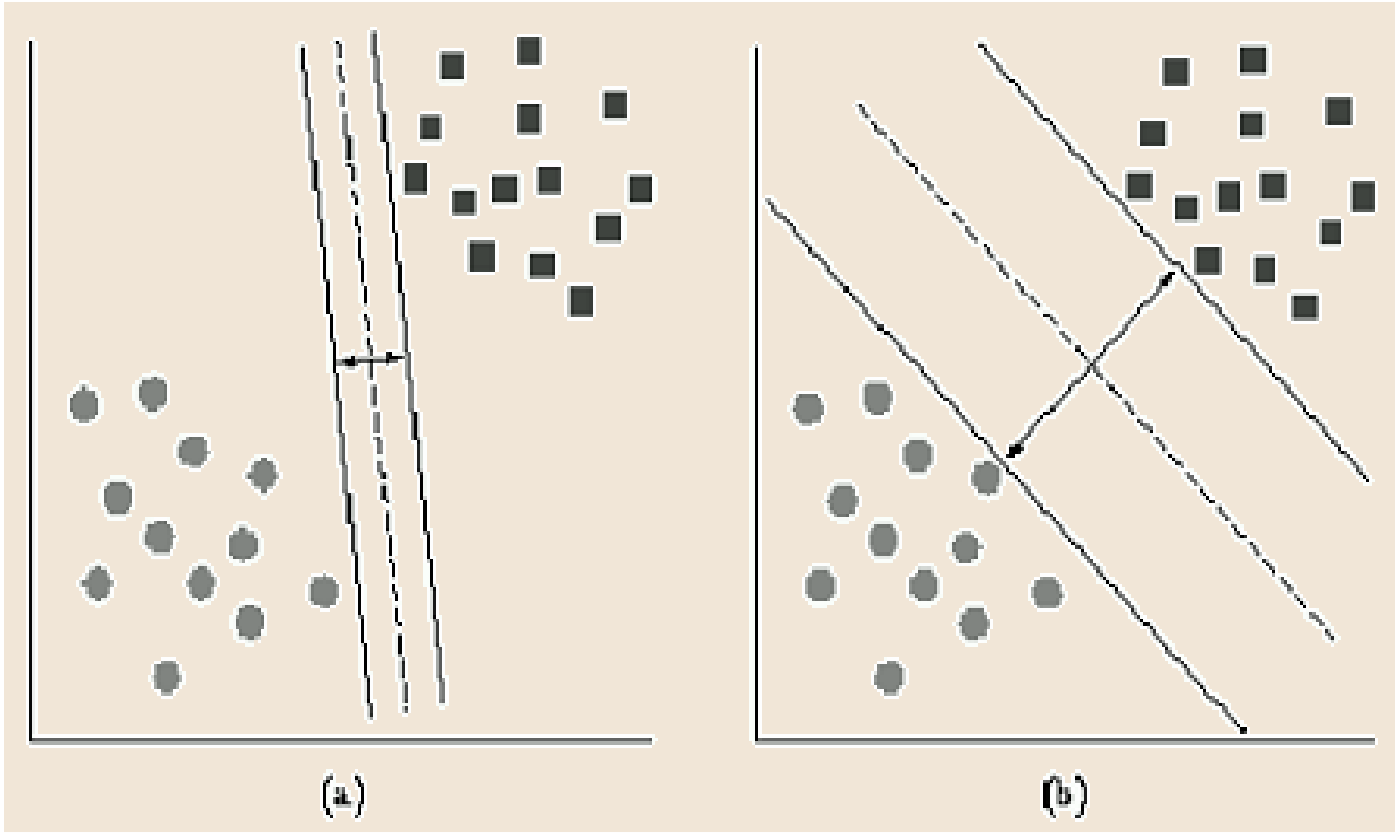
= (5, 0, 3, 0, 1) + 0.5 (2, 1, 2, 0, 0) - 0.25 (1, 0, 0, 0, 2)

= (5.75, 0.50, 4.00, 0.0, 0.5)

基于分类的方法

- Binary text classification: relevant vs. non-relevant
- Adapt a text classifier
  - But classes are extremely unbalanced … because most documents are not relevant
- System maximizes user utility instead of minimizing classification error
  - Because some errors are more important than others
  - 2 credits for a good document, 1 penalty for a bad document

SVM



模式匹配算法

模式匹配算法

- 基本的文本扫描操作-- 字符串搜索:
  - 给定一个单独的模式p (搜索串)和一个输入串s, 如果p 是s 的子串就回答yes，否则回答no
- Brute force串匹配算法
- 快速串匹配算法
  - KMP算法
  - BM算法
- 多模式匹配WM算法

Brute Force算法

- 搜索串(模式): $p_1p_2\ldots p_m$
- 文本串: $s_1s_2\ldots s_n$ (通常n >> m)
- 将模式和m个字符的字串$s_ks_{k+1}\ldots s_{k+m-1}$进行匹配，k从1到n - m +1.
- 模式要么和子串匹配，要么找到一个位置发现二者不匹配

$$p_1 \cdots p_i \cdots p_m$$
$$s_1 \cdots s_k \cdots s_j \cdots s_{k+m-1} \cdots$$

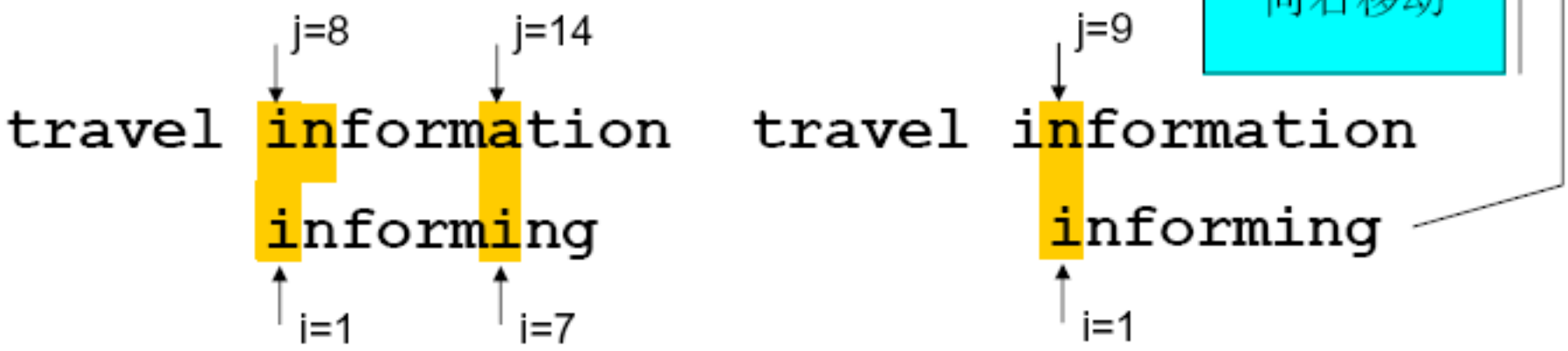
Brute Force算法(2)

- **begin**
  - i := 1
  - j := 1
  - **while** $i \le m$ **and** $j \le n$ **do**
    - **if** $pi = sj$ **then begin** $i := i+1$; $j := j+1$ **end**
    - **else begin** $j := j - i + 2$; i :=1 **end**
  - **if** $i > m$ **then return** "yes **else return** "no"
- **end**

此循环可以更早地终止: $j \le n - m + 1$

最多 $n \times m$ 次

模式向右移动



Brute Force算法(3)

- Brute force算法较慢

Position	1	2	3	4	5	6	7	8
Text String	a	b	d	a	d	e	f	g
Pattern	a	b	d	f				

a b d f ← Brute force: 移动一个字符

a b d f ← 聪明的做法：移动三个字符

- 在位置4出现不匹配的现象，Brute force算法只移动了一位
- 由于前三个位置(a b d)都匹配成功了，移动一个位置不可能找到“a”，因为它已经被识别为“b”
- 在文本的前缀被匹配成功后，你已经对文本串有了一些了解

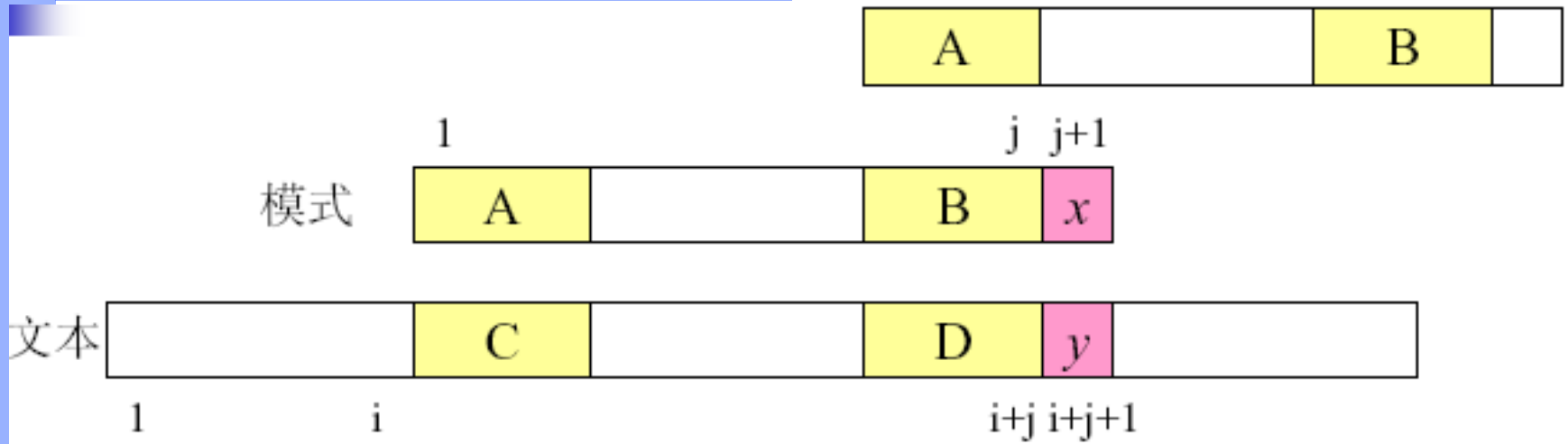
KMP算法 -- 基本思想

- KMP(Knuth-Morris-Pratt) 匹配算法
- 充分利用在一次失败匹配过程中获得的知识，避免重复比较已经知道的字符

位置：	1	2	3	4	5	6	7	8
文本串	a	b	d	a	d	e	f	g
模式	a	b	d	f				

  - 关于文本的知识: a b d ?
  - 需要在文本串中找到另一个“a”
  - 我已经知道“a”不可能出现在下两个字符中，因此可以向右移动三个字符
  - 需要实现对模式(pattern)中的字符进行分析

KMP -- 移动模式



- 从匹配成功的子模式中找出“能够相互匹配的最长的前缀和后缀”
- 通过对模式的分析，我们知道A=B，在匹配过程中知道A=C, B=D;
- 移动模式，让A和D对齐，从位置i+j+1处开始匹配
- 如果在模式[1..*j*]中没有重复模式, 则可以直接移动*j*个字符
- 有重复子模式的模式需要更多的时间去匹配，例如：aaaaaab

KMP算法 -- Shift表

- 匹配失败时，决定移动多少个位置

关键词 字符	匹配的长度 重复子串	跳过的字符数
a	0	1
b	0	1
c	0	2
a	0	3
b	1	3
c	2	3
a	3	3
c	4	3
a	0	8
b	1	8

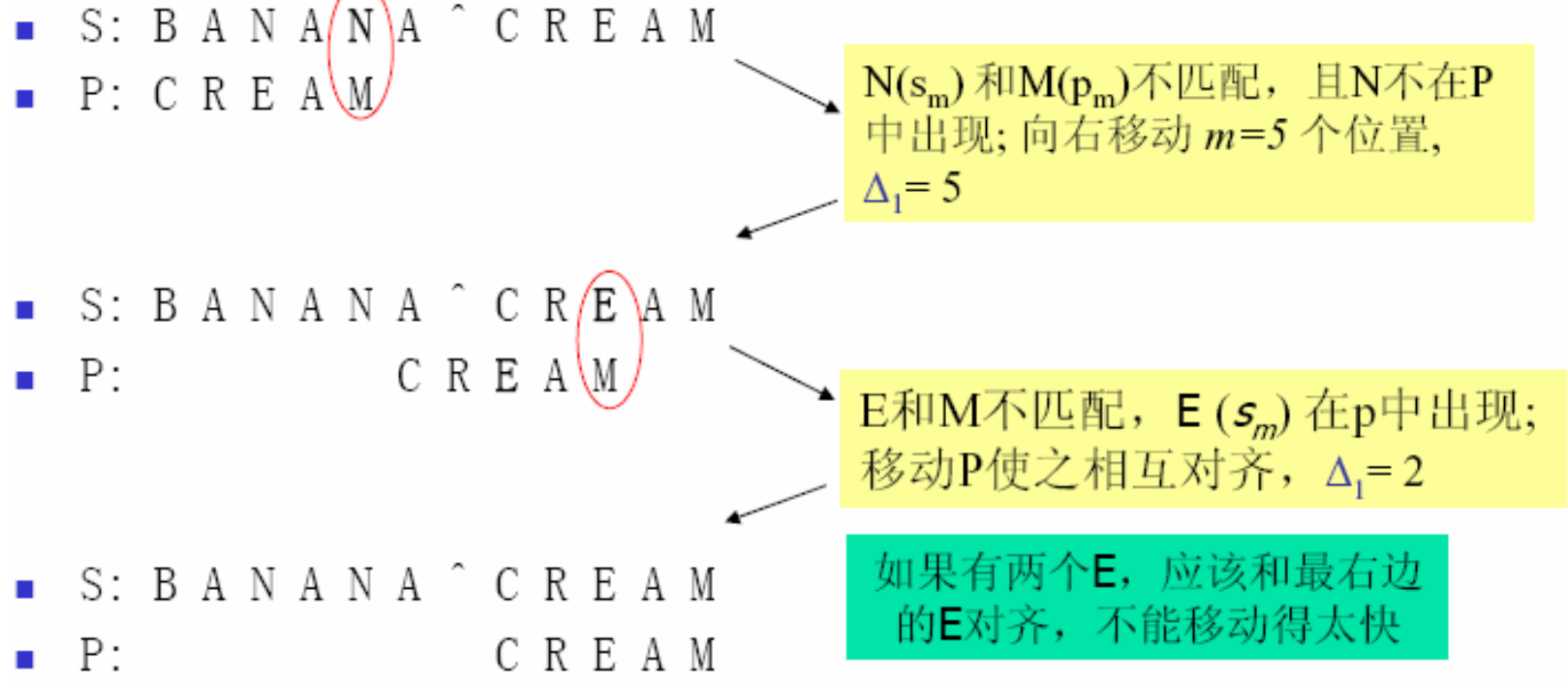
因为
abcab≠abcac
所以，找不到
“重复子串”

KMP算法 -- 性能

- Shift表可以在O(m) 的时间复杂度下，通过对模式的分析而获得
  - m是模式长度
  - 对每个模式字符，需要计算当匹配失效发生时应该跳过几个字符
- KMP算法花费O($m+n$)的时间来解决此问题

BM (Boyer-Moore) 算法

- 基本想法:（模式串）从右到左匹配输入串
- 比较p_m 和s_m，如果s_m不在关键词中出现，那么从s的前m个字符中任何一个字符开始的字符串都不可能和p相比配，因此可以安全地向右滑动m个字符，从而避免m-1次不必要的匹配。



BM算法 -- Δ_1 Shift表

- 如果模式的长度为p 并且字母表的大小为q (对小写字母来说q=26),则需要$p \times q$的Δ_1 shift表

α	1	2	3	4	5
M	1	2	3	4	Φ
A	1	2	3	Φ	1
E	1	2	Φ	1	2
R	1	Φ	1	2	3
C	Φ	1	2	3	4

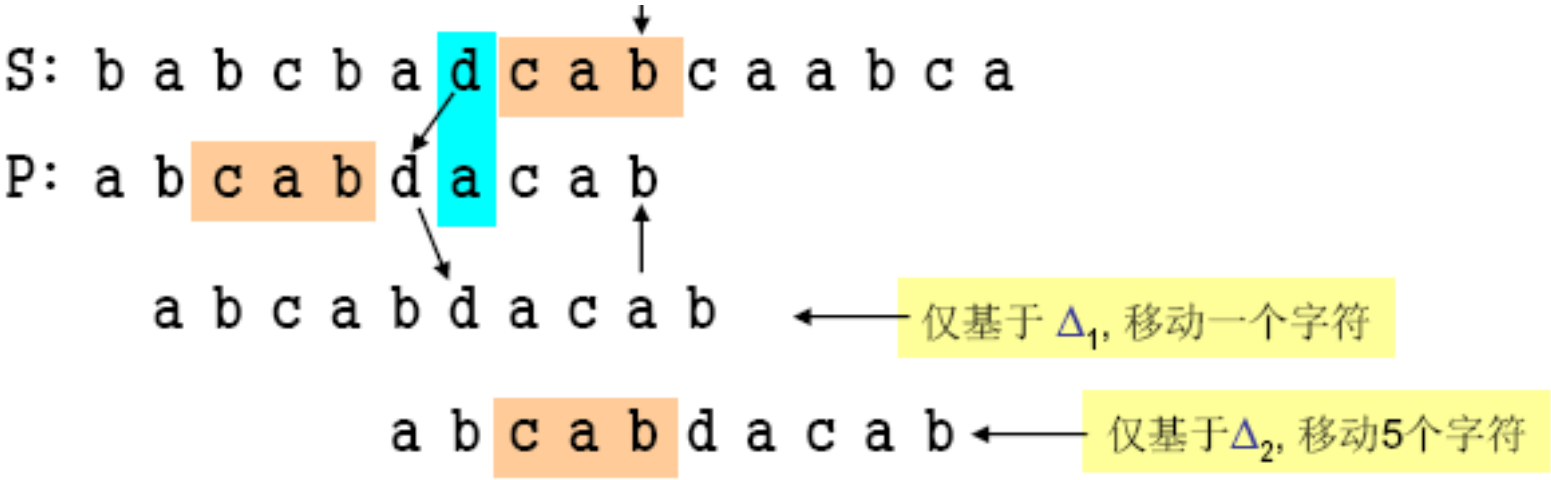
最右不匹配的
字母位置编号

α：字母表 – {C, R, E, A, M}

模式P=CREAM

BM算法 -- Δ_2 Shift表

- p的尾部已经和s的某些子串相匹配，但再向左移动一位就不匹配了，此时使用 Δ_2 Shift表



  - 如果应用 Δ_1, 只能移动一个字符
  - 和KMP相似, 可以知道已经匹配成功的模式后缀cab在模式的前面已经出现过, 因此移动1个字符肯定无法匹配成功
  - 应用移动模式，用前面已经出现的模式模式和S中相应的文本对齐 $\Delta_2 = 5$

BM算法 -- Δ_2 Shift表

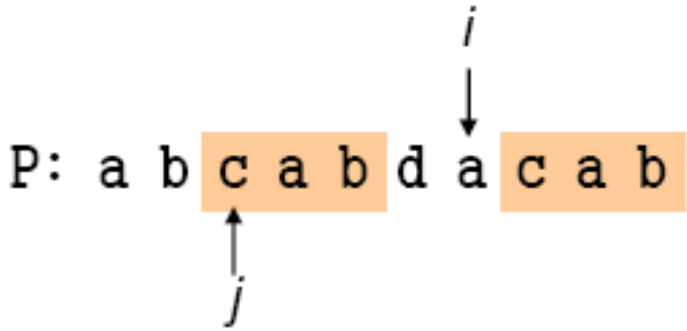
If mismatch at p[i]

len = m - i

find largest j such that

p [j .. (j + len - 1)] = p [(i+1) .. m]

shift2[i] = i – j + 1



BM算法 -- Δ_1和Δ_2

- 同时计算出Δ_1和Δ_2，并使用最大的shift

```
S: b a b c b a d c a b c a a b c a
P: a b c a b c a c a b
                  a b c a b c a c a b    Δ1 = 7
                a b c a b c a c a b      Δ2 = 5
```

  - $\Delta_1 = 7$, 因为不匹配的字符d 不在剩余的模式中出现
  - 如果基于重复子模式来考虑，则$\Delta_2 = 5$
  - 同时计算出Δ1 和Δ2，并使用最大的shift

BM算法 -- 性能

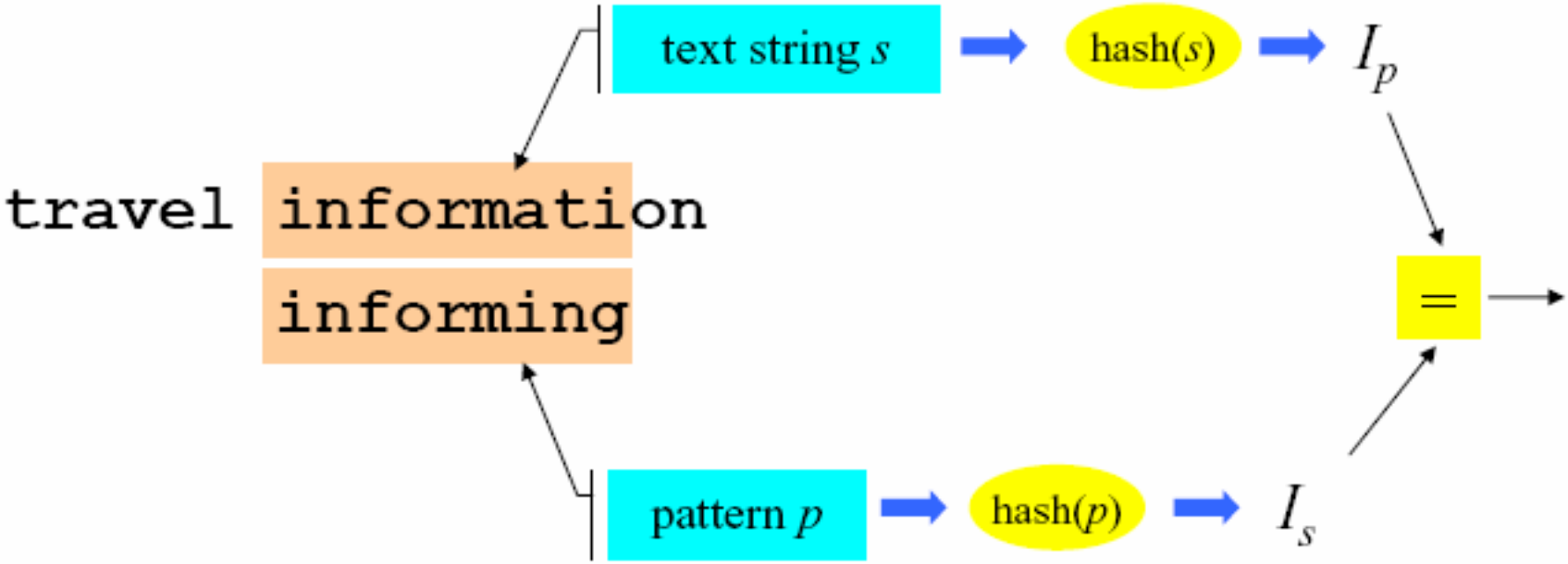
- 在理论上和实践上，BM都比KMP更快
  - 它跳过了输入串中对匹配不可能有贡献的点
    - KMP对文本中的每个字符都进行匹配
    - 一般来说，BM匹配很少的字符
- 当模式中重复部分很少时，效率最高(类似KMP)
- 需要比较的字符数比输入字符串的长度*n*要小的多
- 很难应用于多关键词匹配(KMP也一样)

算法选择: 启发式

- 如果关键词的长度很小(1-3 characters),可以使用Brute force算法
- 如果字母表很大，KMP算法是一个合适的选择
- 否则，特别是对于长文本来说, BM 是最佳选择

Karp-Rabin算法

- Karp-Rabin算法使用一个hash函数来降低将关键词和每一个m字符的字串相对比的开销



WM(多模式匹配)算法

- 1994年， Sun Wu 和Udi Manber 提出的快速的多模式匹配算法通过SHIFT, HASH, PREFIX和POINTER四个表格完成匹配

WM算法 -- Shift表的构建

- m=最短模式的长度，B=被考察的“块”的大小
- 大小：B×字母表大小
- 字母表中每B个字符构成的字符串被散列为一个整数h，h就是SHIFT表的入口
- SHIFT中每一项的值决定在文本中出现某B个字符组成的字符串时pattern的移动距离
- 两种情况
  - X不在任何一个pattern中出现，SHIFT[h] =m-B+1
  - X在某些pattern中出现，且这些pattern中出现X的最右位置是q，则SHIFT[h]=m-q

WM算法 -- Shift表的构建

字符集为{S H E R}

SHEE HERS

对于字符集中任意两个字符的组合，通过一定规则，散列成一个整数**h**，作为其在**SHIFT**表中的入口

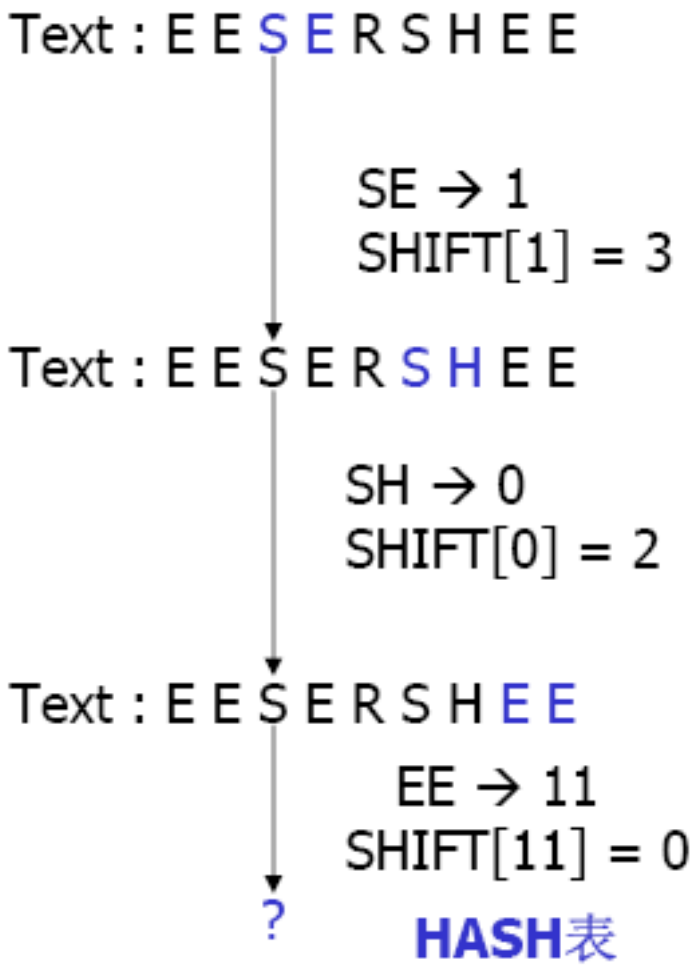
SHIFT表

可能的字符组合	移动距离
SH(0)	2
SE(1)	3
SR(2)	3
SS(3)	3
HS(4)	3
HE(5)	1
HR(6)	3
HH(7)	3
ES(8)	3
EH(9)	3
ER(10)	1
EE(11)	0
RS(12)	0
RH(13)	3
RE(14)	3
RR(15)	3

WM算法 -- Shift表的使用

SHIFT表

可能的字符组合	移动距离
SH(0)	2
SE(1)	3
SR(2)	3
SS(3)	3
HS(4)	3
HE(5)	1
HR(6)	3
HH(7)	3
ES(8)	3
EH(9)	3
ER(10)	1
EE(11)	0
RS(12)	0
RH(13)	3
RE(14)	3
RR(15)	3



WM算法 -- Hash表的构建

- 如当前所考察的字符串在SHIFT表中映射的值为0，这意味着现在考察的text中的子串已经和某个或某些pattern的最后B个字符匹配。
- 但是并不知道是哪个或哪些pattern匹配了，为了避免将text中的子串和每一个pattern比较，采用散列的技术解决这个问题。
- HASH[h]中存放着一个指针，指向POINTER表的入口

WM算法 -- 基本思想

HASH表

可能的字符组合	HASH值
SH(0)	0
SE(1)	0
SR(2)	0
SS(3)	0
HS(4)	0
HE(5)	0
HR(6)	0
HH(7)	0
ES(8)	0
EH(9)	0
ER(10)	0
EE(11)	0
RS(12)	1
RH(13)	2
RE(14)	2
RR(15)	2

POINTER表

0	SHEE
1	HERS

PREFIX表

PATTERN	前缀
SHEE(0)	SH
HERS(1)	HE

Text : E E S E R S H E E

计算前缀**text_prefix = SH**的散列值

EE → h(11)

SHIFT[11] = 0 ⟶ HASH[11] = 0

WM算法 -- 匹配过程

- 1、根据文本当前正考察的B个字符计算其散列值h（从T_{m-B+1}…T_m开始）
- 2、检查SHIFT[h]的值，如果SHIFT[h] > 0，那么移动SHIFT[h]个字符，返回第一步，否则，进入第三步。
- 3、计算文本中对应“前缀”的散列值，记为text_prefix.
- 4、对符合HASH[h] ≤ p < HASH[h+1]的每一个p值，检验是否存在PREFIX[p] = text_prefix. 如果他们相等，就对text和pattern进行直接检查，pattern由POINTER[p]获得。

TREC

TREC

- http://trec.nist.gov/
- 文本检索会议(text retrieval conference,简称TREC)
- 美国国家标准技术局(NIST)和国防部高级研究计划局(DARPA)组织召开的一年一度的国际会议。
- 文本检索领域最权威的国际会议之一。
- TREC6~11 (1997--2002)：文本过滤项目 Filtering Track

TREC Filtering Track

- 文本过滤项目Filtering Track
- 任务定义：
  - 给定一个主题描述（即用户需求）
  - 建立一个能从文本流中自动选择最相关文本的过滤模板（Filtering Profile）
  - 随着文本流的逐渐进入，过滤系统自动地接受或拒绝文本，并得到反馈信息（由TREC的组织评估单位提供），根据反馈信息自适应地修正过滤模板

TREC Filtering Track

- 文本过滤项目包含三个子任务。
  - 分流(Routing)
    - 用户需求固定、训练文本充足、无需给定相关度
  - 批过滤 (Batch Filtering)
    - 用户需求固定、训练文本充足、需要给定相关度
    - 对测试文档集中的全部文本逐一作出接受或拒绝的决策。
  - 自适应过滤 (Adaptive Filtering)
    - 用户需求变化、训练文本很少、不断调整相关度
    - 仅仅从主题描述出发，不提供或只提供很少的训练文档
    - 对输入文本流中的文本逐一判断。
    - 对“接受”的文本，能得到用户的反馈信息，用以自适应地修正过滤模板。而被“拒绝”的文本是不提供反馈信息的。
    - 这是最接近真实环境也是最困难的子任务。

TREC-7（2000）语料库

- 文本过滤的语料库采用了医学文献语料库OHSUMED。
- 它由1987－1991年的医学文摘组成，共含文本近35万篇，总容量为400M。
- 其中87年的文摘将作为训练语料，而88－91年的文摘将作为测试语料。

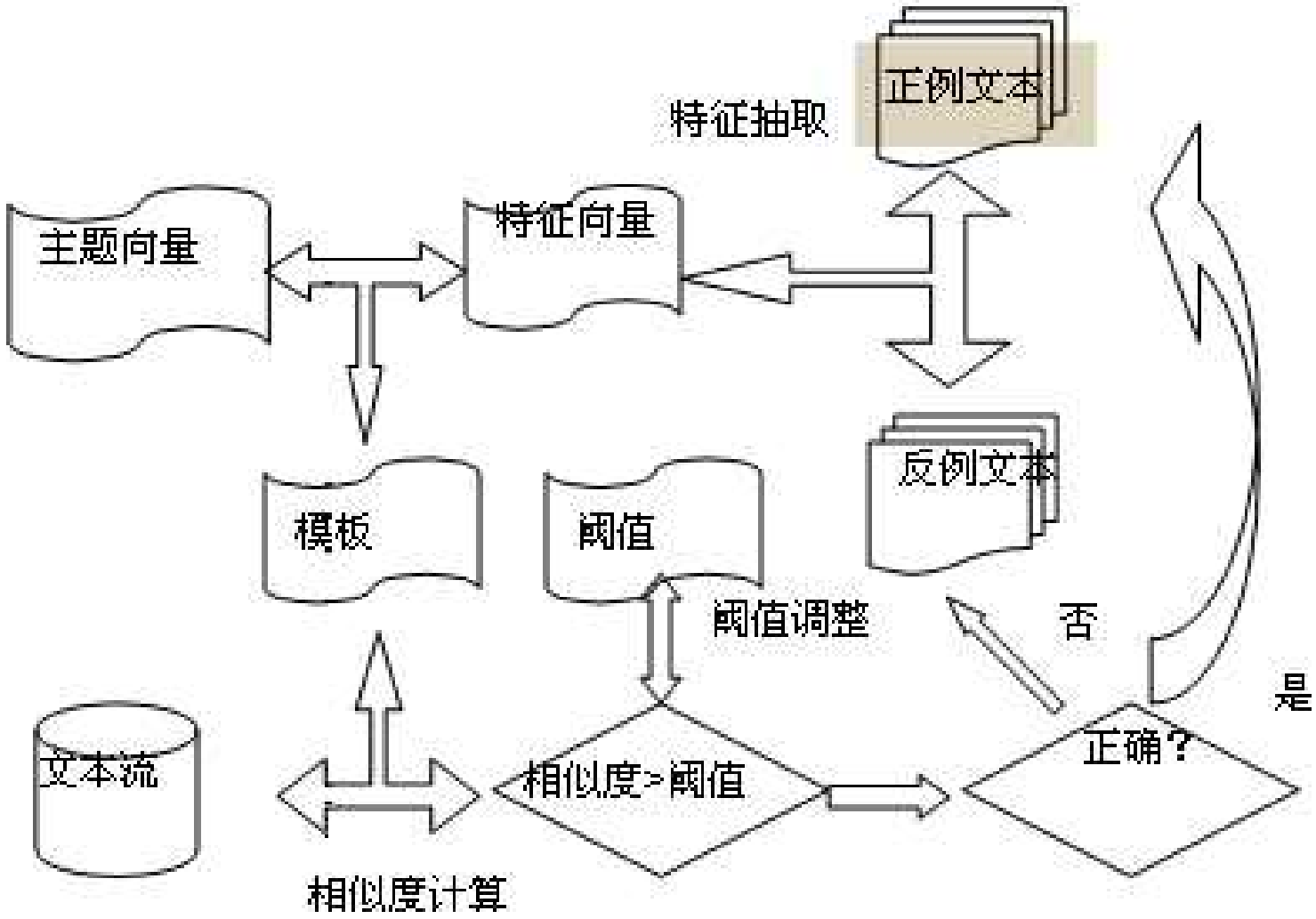
TREC-7（2000）评测

- 用户的信息需求表现为疾病或症状描述，共有63个，由标题和描述两个部分组成，例如：
  - < title > 60 year old menopausal woman without hormone replacement therapy
  - < desc > Are there adverse effects on lipids when progesterone is given with estrogen replacement therapy
  - 标题部分是一些关于病人的介绍
  - 描述部分是专业人员根据病人介绍而人工编写的查询语句

TREC-7（2000）评测

- 对每个主题，TREC都提供了全部训练文本的相关性评价。
- 一共有三个级别的相关性评价：definitely，possibly，not relevant。前两者都表示相关，但程度不同。
- 相关文本在全部文本的比例中是非常稀少的，平均相关文本数为5年61.4篇，即平均10000篇文本中只有1.76篇相关文本。
  - 对于batch filtering，提供训练集中全部的相关文本作为训练数据。
  - 而对于adaptive filtering，只提供了少量的相关文本作为训练数据。
- 由此带来的数据稀疏问题是文本过滤中的难点之一。

文本过滤系统(复旦大学)



文本过滤系统(复旦大学)

- 基于向量空间模型的文本过滤
- 训练算法
  - 初始的模板则是主题向量、正例特征向量和伪正例特征向量的加权和
    - 伪正例：与模板向量高度相似但又不是给定的正例文本的那些文本
    - 抽取若干与相关文本集互信息量最大的词汇作为最优特征

$$Pf_0(Q) = \alpha \cdot P_0(Q) + \beta \cdot P_1(Q) + \gamma \cdot P_2(Q)$$

  - 计算初始向量和全部训练样本之间的相似度,从而为每个主题选择最优的初始相似度阈值

文本过滤系统(复旦大学)

- 自适应算法
  - 阈值的调整

(1) 若 $cor(T)<rtv(T)*20\%$,且 $rtv(T)>\max(M_0,4)$,则 $TH(T+1)=TH(T)*1.2$.

- 即如果准确率过低,而检出的文本又不太少,则迅速提高阈值.

(2) 若 $rtv(T)>M_0$ 且 $RTV(T)>M(T)$,则 $TH(T+1)=TH(T)*1.1$.

- 即如果检出的文本多于必需的,则提高阈值

(3) 若 $rtv(T)<M_0$ 且 $RTV(T)<M(T)$,则 $TH(T+1)=TH(T)*0.9$.

- 即如果检出的文本少于必需的,则降低阈值.

$cor(T)$:在 T 时段正确检出的文本数

$rtv(T)$:在 T 时段检出的文本数

$COR(T)$:到 T 时段为止正确检出的文本数

$RTV(T)$:到 T 时段为止检出的文本数

$M(T)$:到 T 时段为止必须检出的文本数

$TH(T)$:T 时段的相似度阈值

文本过滤系统(复旦大学)

- 模板的修改

$$Pf'(Q) = \alpha' \cdot P_0(Q) + \beta' \cdot P_1(Q) + \gamma' \cdot P_3(Q).$$

$P_3(Q)$是由反例文本抽取的特征向量

$$P_3(Q) = (p_{31}, p_{32}, \cdots, p_{3W})$$

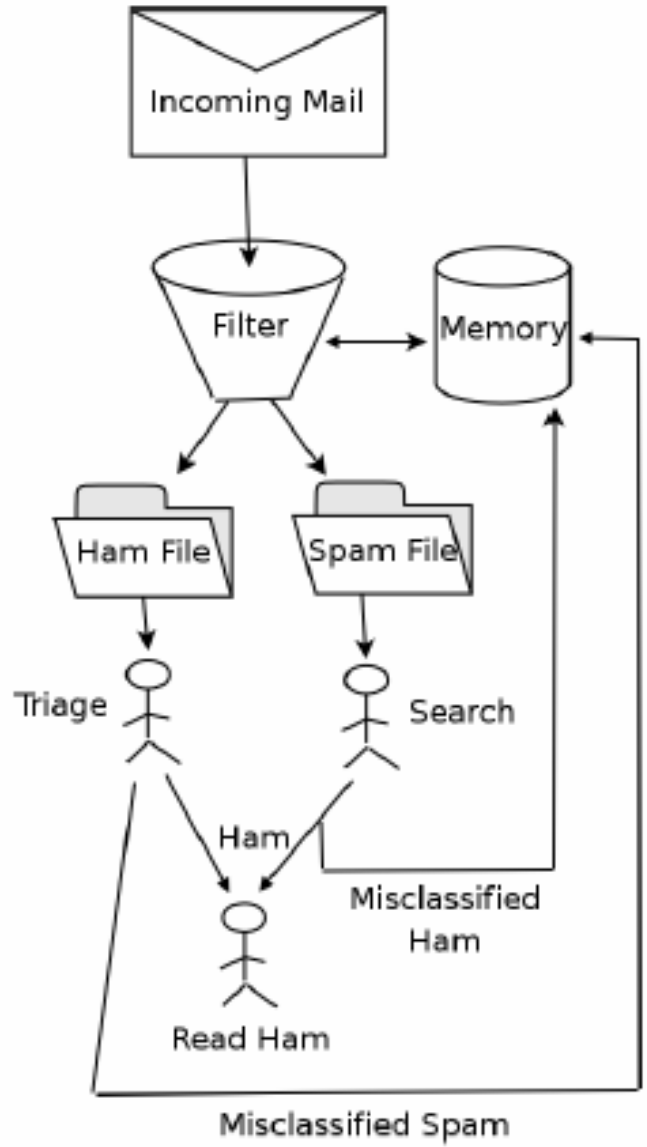
$$p_{3i} = \begin{cases} \log MI(w_i, irrel(Q)), \text{if } \log MI(w_i, irrel(Q)) \geq 3 \\ 0, \qquad \text{otherwise} \end{cases}.$$

$irrel(Q)$表示主题 Q 的反例文本.

邮件过滤

垃圾邮件的定义

- 垃圾邮件是指
  - 向未主动请求的用户发送的电子邮件如广告、刊物或其他资料;
  - 或没有明确的退信方法、发信人、回信地址等的邮件;
  - 或利用网络从事违反网络服务供应商的安全策略或服务条款的行为和其他预计会导致投诉的邮件。



公开语料

- PU
  - http://www.aueb.gr/users/ion/publications.html
  - 保留标题和正文的纯文本，词汇用整数代替
  - bare、lemm、stop、lemm_stop
- Ling Spam
  - http://www.aueb.gr/users/ion/publications.html
- Spam Assassin
  - http://www.spamassassin.org/publiccorpus/
- Spambase
  - 表示成向量形式
- Enron-Spam
  - http://www-2.cs.cmu.edu/~enron/
- Trec
  - Trec 05 -- Trec 2007
  - Trec 06 加入了中文语料

公开语料

语料名称	Ham	Spam	total
PU1	618	481	1099
PU2	579	142	721
PU3	2313	1826	4139
PUA	571	571	1142
Ling Spam	2412	481	2893
Spam Assassin	4150	1897	6047
Spambase	2788	1813	4601
Enron			619,446

评价体系

- 正确率(Precision, TPR)
  - tp / (tp + fp)
- 召回率(Recall)
  - tp / (tp + fn)
- 精确率(Accuracy)
  - (tp + tn) / N
- 错误率
  - (fp + fn) / N
- 代价因子TCR[2]
  - TCR = N_s / (λ fp+fn)

实际 / 判定	Spam	Ham
Spam	tp	fp
Ham	fn	tn

N = tp + fn + fp + tn

N_h = tn + fp

N_s = tp + fn

评价体系

- Ham错误率(hm%)
  - hm% = fp / (fp + tn)
- Spam错误率(sm%)
  - sm% = fn / (fn + tp)
- 对数平均错误率(lam%)
  - $lam\% = logit^{-1}((logit(hm\%) + logit(sm\%))/2)$
  - $logit(x) = log(x/(1-x))$
- ROC曲线

Bogofilter过滤方法

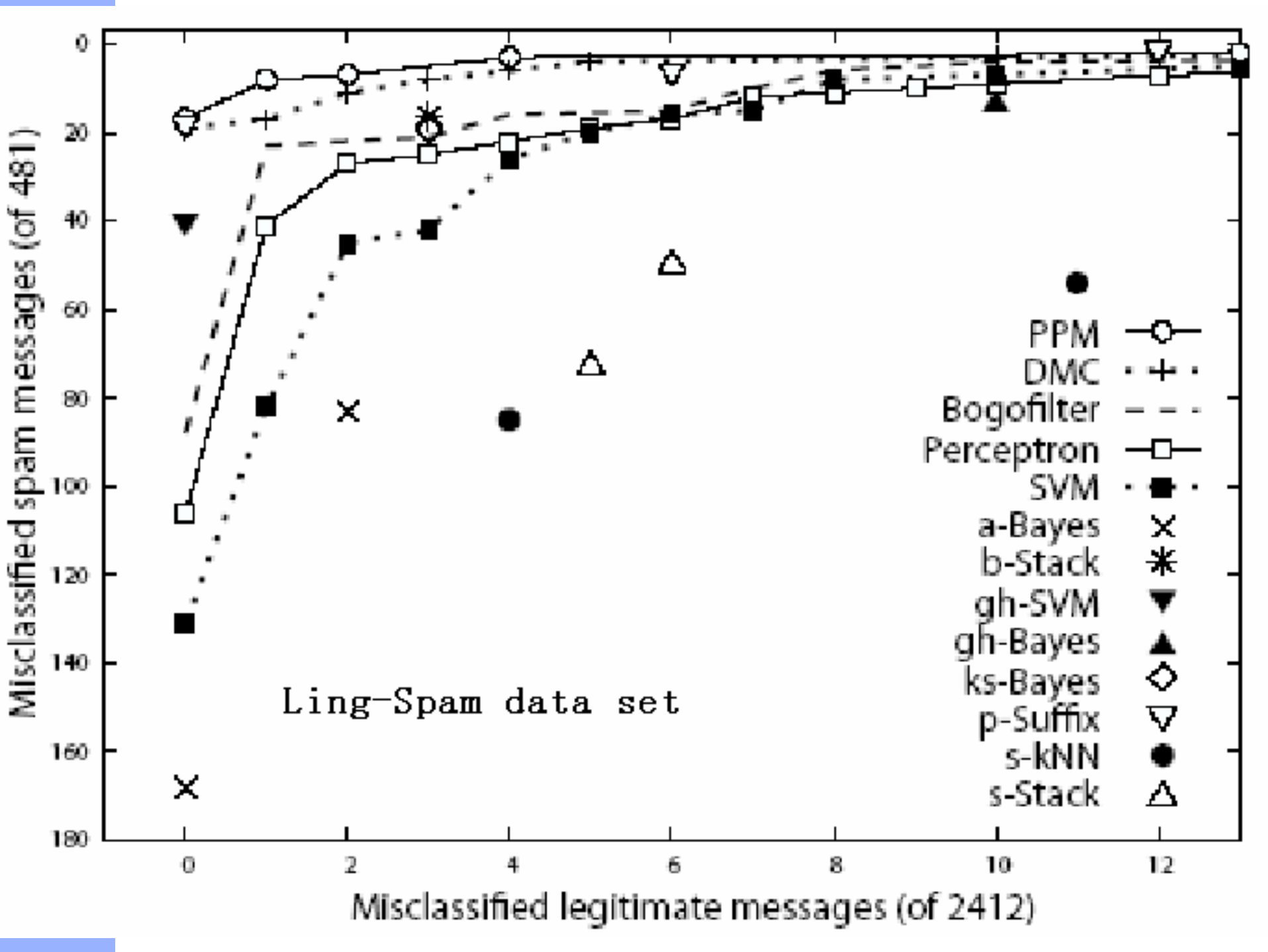
- 基于Bayes原理；
- 自动分析邮件文本或者标准输入文本，基于设定的正常或者垃圾邮件信息判断邮件属性，并返回邮件是否属于垃圾邮件；
- 通过对邮件的头（header）和内容（body）进行统计分析来分类，并能通过用户的分类和纠正来学习；
- 对附件名进行统计，但是忽略附件内容，如图片等 。

http://bogofilter.sourceforge.net/

DMC过滤方法简介

- 采用统计数据压缩技术;
- 将邮件作为字节流，无需常规特征提取的步骤（Tokenization, etc）;
- 可以提取字符或二进制流层次上的特征做为过滤标准;
- 具有快速判别和动态更新的特点;
- 实际过滤结果明显优于bogofilter

Andrej Bratko, Gordon V. Cormack, Bogdan Filipic, Thomas R. Lynam and Blaz Zupan，Spam Filtering Using Statistical Data Compression Models，Journal of Machine Learning Research，Dec. 2006. pp 2673--2698


Misclassified spam messages (of 481)
Misclassified legitimate messages (of 2412)
Ling-Spam data set
PPM
DMC
Bogofilter
Perceptron
SVM
a-Bayes
b-Stack
gh-SVM
gh-Bayes
ks-Bayes
p-Suffix
s-kNN
s-Stack
0
20
40
60
80
100
120
140
160
180
0
2
4
6
8
10
12

垃圾邮件—文字变形

文章指出，中 共政.治局这个动作说明，胡.錦.濤和江-澤-泯无法妥协，政.治谈判已无法打破僵局，胡锦涛被迫出手將陈良宇拉下马，预示着双方已几乎"摊牌"。

港评：胡.錦.濤驾驭能力胜江-澤-泯

香港一项评-论-认为，胡锦涛这次向陈良宇开刀，并未依靠什么元老的力量，足见胡.錦.濤在驾驭政局和对官场的掌控能力不会比江-澤-泯低。

评-论-又说，江-澤-泯在一九八九年"六- 四"事件后，从上海市委书记升任总书记，为了巩固权.力，利用当时的中-共元老邓小平、陈云、李先念等人的力量，先后將对自己不服气的陈希同以贪腐罪名送入监牢，將另一名对自己不服气的政.治局委员杨白冰的武功废掉。然而，作为江-澤-泯的亲信陈良宇今天却因贪腐弊案而下台，一饮一啄，因果相报。

专家评-论-：制度问题，无法挽回中-共解体命运

时事评-论-家陈劲松指出，陈良宇固然贪腐，但恐怕连大"老虎"都算不上。谁都知道，日益深重的中 共官场腐败，起源于一党独.裁的政.治制度。只有建立起互相监督、彼此制衡的 机.制，包括新聞 自`由和司法独立，腐败才可能从根本上和最-大程度上得到遏制。但不论是江，还是胡，都显然无心于此。

大 J1 元专栏作家龙延指出，最近，中纪委等部门组织党员领导观看中-共摄制的"苏共亡党历史教训"教育片，告诫中共党员拒腐防变。江系人马陈良宇的下台不仅是胡为了巩固权.力，更重要的是，在《九"評共 产_党》和退'党大潮的强烈冲击下，为了化解亡党危机，中共有意要在腐败问题上下一些功夫。他表示，历史证明，中 共的邪恶流(亡民)本性是不可能改变的，其被解体也是历史的必然。单就反腐败本身是有正面意义的，但是想以反腐来挽救中-共被解体的命运，是逆天意的，也是做不到的。

垃圾邮件—干扰文字

OEM software - throw packing case, leave CD, use electronic manuals.
Pay for software only and save 75-90%!

Find incredible discounts! See our special offers!
TODAY TOP 10 ITEMS

$49 Windows XP Pro w/SP2
$79 MS Office Enterprise 2007
$79 Adobe Acrobat 8 Pro
$79 Microsoft Windows Vista Business
$99 Macromedia Studio 8
$59 Adobe Premiere 2.0
$59 Corel Grafix Suite X3
$59 Adobe Illustrator CS2
$129 Autodesk Autocad 2007
$149 Adobe Creative Suite 2
and 1000s more under $49.
http://ah.febhappyad.eu/?93194eAKRH87354TPAUh53889oeulK80577ooadU14036zUkVx
---
IRQ line. You should use whichever one works.
Line 6:
probably continue to increase until more people discover and accept # The rule action could be to drop the packet, to forward the packet, orto tell when it should start to write or read.
config kernel_name root on root_device

垃圾信息

干扰文字

小结

- 文本过滤的概念
- 文本过滤的基本方法
  - 基于查询的方法
  - 基于分类的方法
  - 模式匹配算法
    - Brute force串匹配算法
    - 快速串匹配算法：KMP算法、BM算法
    - 多模式匹配WM算法
- TREC Filtering Track
- 邮件过滤

Any Question?