第八章　信息内容安全与对抗

8.1 中文主动干扰概念和方法

8.2 抗中文主动干扰柔性中文处理算法

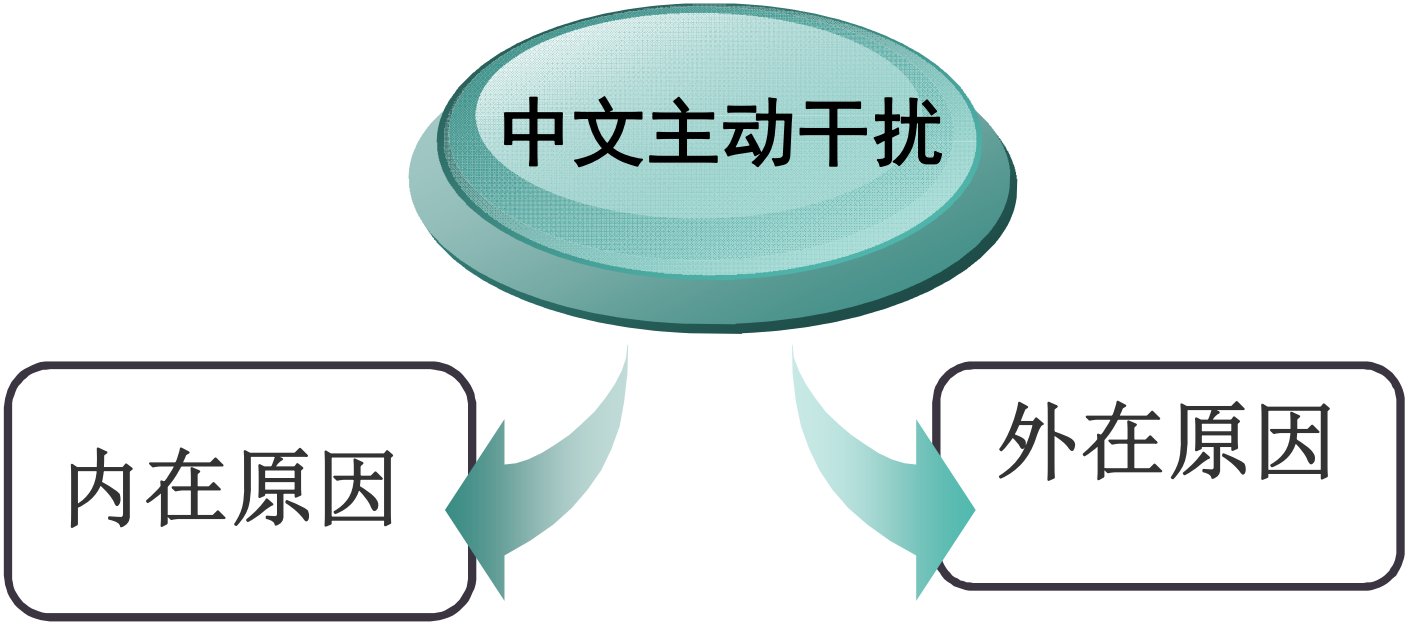
8.3 基于粗糙集与贝叶斯决策不良网页过滤算法

8.4 互联网舆情检测分析系统实例介绍

8.1 中文主动干扰概念和方法

网络信息内容安全攻防双方斗争本土化、形式严峻是信息内容安全与对抗的根本原因。所谓信息内容安全本土化，是指网络攻防双方都以中文为信息处理背景，都熟悉和掌握中文信息处理技术和规律。对于采用本土化的中文不良信息，现有的中文网络搜索软件、过滤软件以及其他相关处理软件系统在处理上是困难的或无效的。

8.1.1 中文主动干扰原因



1. 内在原因

- 中文语言特性不利于自动信息处理。由于汉语的词缺乏形态变化，且不同语言单位（语素、词、短语、句子乃至篇章）之间的界限不清，包括中文分词的困难性，造成中文文本自动处理一直是中文信息处理最困难的工作之一。中文自动分词算法的难点一是歧义切分字段处理，二是未登录词辨识。自1992年《信息处理用现代汉语分词规范》（GB/T13715-92，以下简称《分词规范》）公布和推行后，为中文信息处理中汉语的词汇平面构成了重要支撑平台。
- 信息处理过程中的无意行为。很明显，信息处理过程中，人的行为是可能存在一定比率的无意识出错行为的。例如，在中文写作中存在错别字，就有可能导致与错别字相关的关键词成为不良信息。而在中文输入过程中经常出现的串行、漏行现象，也会造成相关文档出现不良信息。

2. 外在原因

- 政治斗争需要。境内外敌对势力依托互联网，采用主动干扰方法，源源不断地制作和传播大量本应受到严格管制的有害信息和不良信息，将互联网演变为对我进行西化、分化的新“阵地”，导致网上出现大量遭受过主动干扰的中文不良信息。
- 经济利益驱使。搜索引擎优化师SEO为了提高搜索引擎的效率、网上营销商为了给自己的商铺带来巨大的经济利益，这些需求驱使众多的网络技术人员和信息技术爱好者成为网络中文主动干扰信息的制造者，导致网络上出现大量遭受中文主动干扰过的信息。

8.1.2 中文主动干扰概念

中文主动干扰：

网络攻击者了解中文特点，依据汉语同音字、繁体字与简体字并存的特点，利用中文分词技术的困难性，采用在中文连续文本中随机夹杂符号（如宣扬邪教的信息“法？//*轮*！功”），和/或用繁体字/同音字代替（如用“法轮攻”代替“法轮功”）某个中文关键词的方法，欺骗并绕开各种过滤器，造成网络内容安全处理效果大幅下降。

定义1 基本元素

设$\sum$是Unicode字符表集合；E 是英文字母集合（包括大小写），$\mathrm{E}=\{a,b,\cdots,Y;Z\}$，C是Unicode汉字集合；$C=\{$啊，阿，…，鼾，齄$\}$；S是字符、数字、日文假名等字符集合 $S=\{1,2,\cdots,\},\}$，$(E\cup C\cup S)\subset\Sigma$。

8.1.2 中文主动干扰概念

定义2 文本

给定字符表集 Σ，文本（也称文本串）T是信息对的有序链表，记作 $T=\{(t_1,o_1),(t_2,o_2),\cdots,(t_N,o_N)\}$，其中 t_i 是文本串T的第i个元素值，且 $t_i\in\Sigma$，o_i 是元素 t_i 对应的串信息值（1≤i≤N）。文本串T的长度记作 $|T|$，即 $|T|=N$ 对于任意的1≤i≤j≤N，$T[i,j]=\{(t_i,o_i),\cdots,(t_j,o_j)\}$ 称为文本串T的子串，也称为q-gram，其中 $q=j-i+1$ 。文本串T所有的N-q+1个q-gram可以通过在文本串T上每次移动一个大小为q的窗口来获得。

8.1.2 中文主动干扰概念

定义3 中文主动干扰

在不改变文本信息语义的情况下，对文本信息进行干扰，造成计算机无法执行自动中文信息处理的技术。由于删除操作会导致显著的语义改变，故中文主动干扰方法主要采用插入干扰和替代干扰两种方式，插入干扰用函数Ins (·)表示，替代干扰用函数Sub (·）表示。

定义4 插入干扰

$\mathrm{Ins}(\mathrm{t_i}) = \mathrm{T[x,h]}$表示在文本T的第i个元素值后插入$q = \eta - \xi + 1$的子串，$t_k \in (E \cup S)$，$\xi \le \mathrm{k} \le \eta$，其中子串对应的串信息值为零，即$\{o_\xi, o_{\xi+1}, \cdots, o_\eta\} = \varnothing$。

8.1.2 中文主动干扰概念

定义5 替代干扰

$\mathrm{Sub}\left(T[i,j]\right)=T[\phi,\varphi]$ 表示将文本T的子串$T[i,j]$替代为子串$T[\phi,\varphi]$，$t_k \in (E \cup C)$，$\varphi \le k \le \varphi$，其中，子串$T[i,j]$和子串$T[\phi,\varphi]$串信息相等，即$\{o_i, o_{i+1}, \cdots, o_j\} = \{o_\phi, o_{\phi+1}, \cdots, o_\varphi\}$

替代有三种方式：同音替代、同形替代和同义替代，替代的最细粒度为单个汉字，最粗粒度是整个文本。

定义6 干扰相关系数

干扰相关系数记为 $Co = \{type, position\}$，其中 $type = \{T_1, T_2, T_3, T_4\}$ T_1 为插入字符，T_2为同音替代，T_3为同形替代，T_4为同义替代；position为介于0和文本长度$|T|$=N之间的随机数，即0≤position≤N。

8.1.2 中文主动干扰概念

定义7 干信比

令输入文本词数为 N_{in}，输出文本词数为 N_{out}，输入文本长度为N，伪随机序列发生器产生的伪随机序列个数为 N_{ja}，其中 $N_{ja}=\gamma * N$ ，γ 是干扰因子，$0\leq\gamma\leq1$，文本干信比C_{JSR}定义如下：

$$C_{JSR}=\frac{N_{out}}{N_{in}+N_{Ja}}=\frac{N_{out}}{N_{in}+\gamma * N_{in}}=\frac{N_{out}}{N_{in}*(1+\gamma)}$$

8.1.3 中文主动干扰方法

- 中文主动干扰方法包括盲干扰算法和对准干扰算法，其区别在于盲干扰算法产生干扰信息的位置、类型都是随机的，干扰过程中由伪随机序列控制；而对准干扰算法产生干扰信息的位置是确定的，例如以关键词为干扰对象，干扰全文的关键词，干扰类型由伪随机序列控制。

1. 盲干扰算法

盲干扰产生干扰信息的位置和类型都由伪随机序列控制，输入文本T_{in}。 $\mathrm{X}=[c_1 c_2 \cdots c_n]=[w_1 w_2 \cdots w_{N_{in}}]$ ，$c_i \in \Sigma$，为词集合。输出文本 T_{out} 。其中 $Y=[c'_1 c'_2 \cdots c'_p]=[w'_1 w'_2 \cdots w'_{N_{out}}]$ 同理 $c'_i \in \Sigma$ ，$w'_j \in W$，W 为词集合。

盲干扰算法随机产生 $Co=\{type, position\}$ ，根据 $type$ 和 $position$ 决定在文本的何位置进行何种类型的干扰。例如：由伪随机序列随机产生 $Co=\{\mathrm{T}_2, 32\}$ ，则从文本头开始，将第32个词进行T_2型干扰，即替换为该词的同音词，如拼音。

1. 盲干扰算法

伪代码:

Algorithm 1: BJA（ ）//Blind Jamming Algorithm
Input: T_{in}
Output: T_{out}
Initialize Parameter;
Input=T_{in} ;
//create the disturb index by random function
TEXT[i]=PreSegment（T_{in}）;Dis_Index=Random（）;
Begin
WHILE（Dis_Num<INTENSION）
{　//creat the disturb mode by random function
Dis_Mode=Random（）;
Disturb（TEXT[i],Dis_Mode）;Dis_Num++;
}
T_{out}=Revert the disturbed Text using TEXT[i];
End

2. 对准干扰算法

对准干扰是对关键词库中的词进行干扰，干扰信息的类型由伪随机序列控制，关键词库可以是极性词库、不良关键词库或其他。对准干扰算法实例运算如下：指定待干扰关键词，然后编程搜索该关键词在文本中的位置 $position$ (可能搜出一个或多个位置，与文本中该关键词出现次数对应），对于每一个 $position$ 由伪随机序列随机产生一个干扰类型 $type$，从而形成对准干扰 $Co=\{T_2,32\}$ 。

2. 对准干扰算法

伪代码:

Algorithm 2: PJA（ ） //precision jamming algorithm
Input: Tin ,sensitive keyword list
Output: Tout
Initialize Parameter;
Input= Tin;
//create the disturb index by random function
TEXT[i]=PreSegment（Tin）;Dis_Index=Random（）;
Begin
WHILE（Dis_Num<INTENSION）{
//fread the sensitive keyword list,and distinguish this word
If （TEXT[i] in sensitive keyword list） then Dis_Flag=true;
If（Dis_Flag） { //creat the disturb mode by random function
Dis_Mode=Random（）;
Disturb（TEXT[i],Dis_Mode）;Dis_Num++; }}
Tout=Revert the disturbed Text using TEXT[i];
End

8.1.4 中文主动干扰效果评估

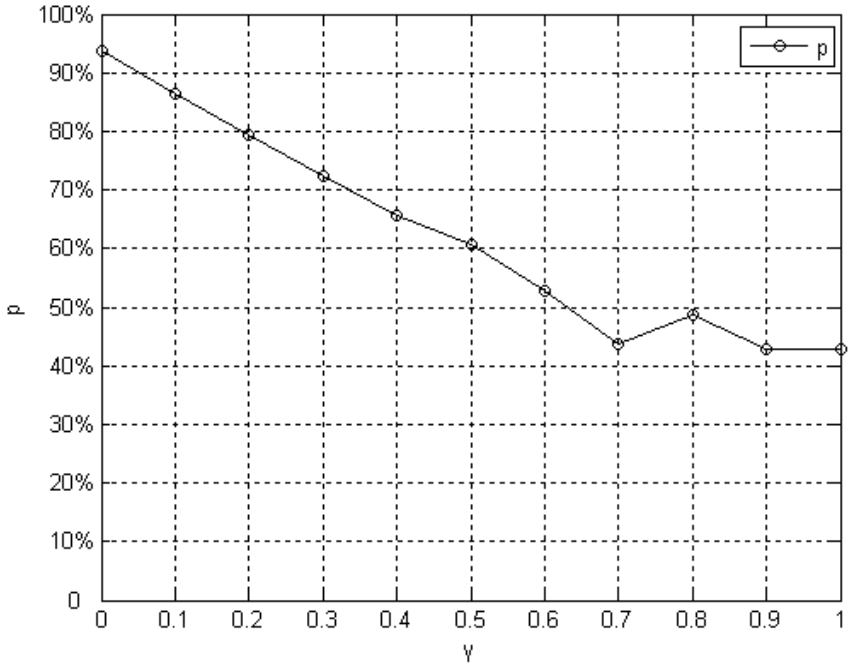
1.中文分词测试

中文信息处理只要涉及句法、语义（如检索、翻译、文摘、校对等应用），就需要以词为基本单位．句法分析、语句理解、自动文摘、自动分类和机器翻译等，更是少不了词的详细信息，因此正确的中文分词是进行中文文本处理的必要条件。

试验采用容量达8.42MB的《人民日报》（1998年1月）公开语料库，整理成3183个TXT文本文件，称为语料库1，记为 $corpus_1$。分词软件采用中国科学院汉语词法分析系统ICTCLAS。

1. 中文分词测试

盲干扰条件下 γ 与 p 关系图如下：



从上图可以看出，ICTCLAS分词系统的分词精度随着干扰强度因子的增大而减小：γ =0时，即不干扰时，分词精度达到93.67%；γ =0.7时，分词精度下降到43.65%；γ ≥0.7时，分词精度趋于稳定，在43%左右。从上图可以看到，当 γ =0.8时，分词精度有缓和上升的趋势，这是因为本实验是在盲干扰条件下进行的，产生的干扰信息类型和位置由伪随机函数控制，当产生的干扰信息类型为 T_1 ，位置在词前或词后时，不影响分词精度。

2. 主题判断测试

对文本按主题进行分类可以提高用户查找效率，文本分类技术已在搜索引擎、数字图书馆技术、信息过滤、信息检索、信息监控等领域中得到了广泛应用。本实验采用简单的向量空间模型对文本进行主题值计算。

主题判别测试语料库有两个，干扰前语料库记为，建库方法如下：采用中文分词实验处理结果文本，对其中的3183个TXT文本文件进行中文分词后，其结果由一组独立的词组成，每篇文本去除停用词后，对剩下的词按频率从高到低进行排序，取前十个高频词作为特征项，对其进行权重赋值并归一化，按下式求出特征项的主题值：

$$v_{ik} = tf_{ik} * tw_{ik}$$

其中，v_{ik}代表特征项在文本d_k中的主题值，tf_{ik}代表特征项t_i在文本d_k中出现的频率，tw_{ik}代表特征项t_i的权重，权重计算采用TFIDF的特征权重计算方法。

2. 主题判断测试

干扰后语料库记为 $corpus_2$，计算 $corpus_1$ 中的特征项在 $corpus_2$ 中的词频，并按下式求出干扰后特征项的主题值：

$$v_{ik}' = tf_{ik}' * tw_{ik}$$

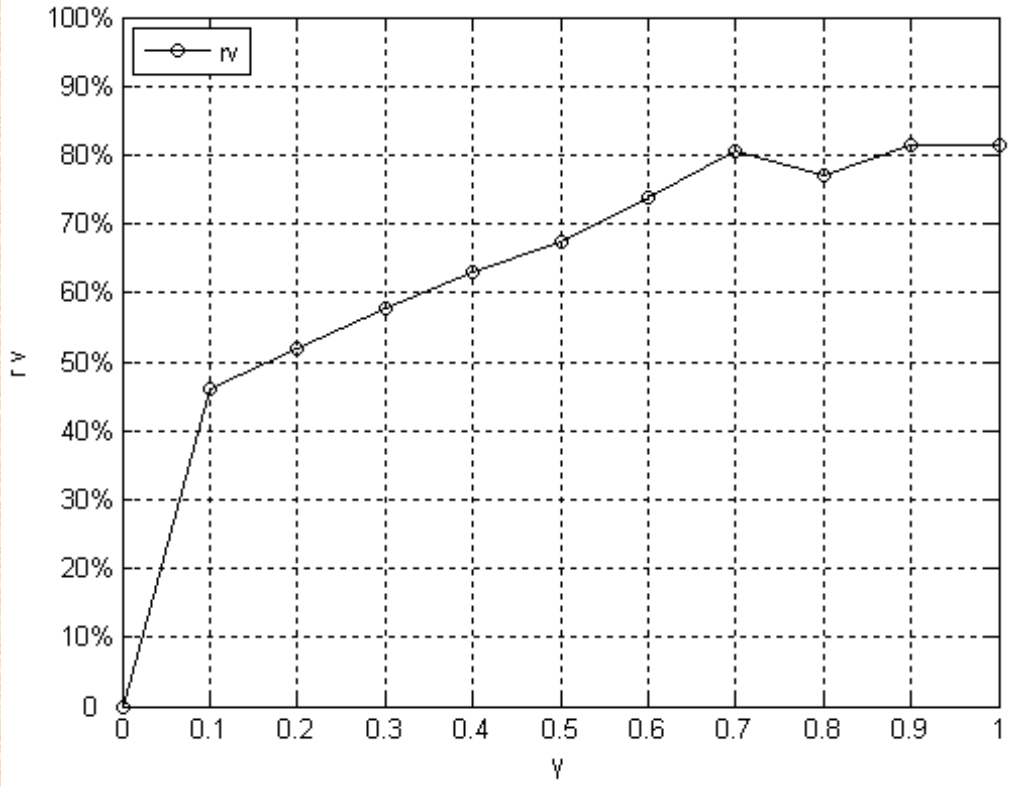
按下式计算主题差值比 ：

$$r_v = \frac{\sum_{k=1}^{TN}\sum_{i=1}^{10}\frac{tf_{ik} * tw_{ik} - tf_{ik}' * tw_{ik}}{tf_{ik} * tw_{ik}}}{TN}$$

其中，v_{ik}' 代表特征项 t_i 在文本 d_k 中的主题值，tf_{ik}' 代表特征项 t_i 在文本中出现的频率，TN 代表测试文本总数。

2. 主题判断测试

盲干扰条件下 γ 与 r_v 关系图如下：



主题差值比 r_v 随着干扰强度因子的增大而增大：

γ =0时，即没有干扰时，主题差值比 r_v 为0；

γ =0.1时，r_v 迅速上升达到45.9%，可见干扰后特征项词频改变很大，导致文本主题改变很大；

γ =0.7时，r_v 缓和上升到80.5%；$\gamma \geq 0.7$ 时，r_v 趋于稳定，在81%左右。可以看到，采用中文盲干扰实验结果作为测试文本，当 γ =0.8时，主题差值比有缓和下降的趋势，这与前一个实验结论一致。

3. 文本倾向性测试

本实验语料库采用由中科院计算所研究人员整理的关于酒店评论的语料库，记为 $corpus_3$，为使本实验数据更具有说服力，本实验情感倾向性分析软件仍采用由同一研制的情感分析系统Sentifier。

本实验的关键词词库由褒义词词库和贬义词词库组成。词库来自于《褒义词词典》和《贬义词词典》，通过删除一些不常用的词语，添加一些近年出现的新词，共同构成关键词词库。

3. 文本倾向性测试

实验结果如下表所示：

中文主动干扰下Sentifier准确率表

	消极文本		准确率	积极文本		准确率	综合准确率
	（neg：2999）			（pos：3000）			
干扰前	neg:2259	pos:640	78.63%	pos:2292	neg:708	76.40%	77.52%
γ =0.1盲干扰	neg:2365	pos:634	78.83%	pos:2246	neg:754	74.86%	76.85%
γ =0.5盲干扰	neg:2289	pos:709	76.30%	pos:1850	neg:1150	61.67%	68.99%
γ =1.0盲干扰	neg:2415	pos:583	80.50%	pos:1125	neg:1875	37.50%	59%
对准干扰	neg:2561	pos:438	85.36%	pos:1542	neg:1458	51.40%	58.38%

8.2 抗中文主动干扰柔性中文处理算法

中文网络内容安全改进措施包括两个方面：

a) 改进现有的中文文本自动处理算法，提高自学习能力，包括如何尽可能地缩维而不改变原来的表达集合，如何实现正交的基于布尔模型的OCAT规则推理等；

b) 提供更有效的过滤不良网页/文本的算法，包括根据字频同现（关联特征）实现关键字符串提取、串匹配及其在网络内容分析中的应用等。

8.2.1 柔性中文串匹配算法

1. 中文串匹配的形式化定义

（1）记号

- C：全部简体汉字和繁体汉字集合
- U：包括汉字及其他字符的Unicode集合，
- Z：整数集合
- p：用于进行匹配的中文字符串，
- t：待匹配的纯中文文本，
- v：待匹配的夹杂中文文本，

1. 中文串匹配的形式化定义

（2）形式化定义

中文串匹配形式化定义包括以下算法：

- Lenth(x)：该算法为统计函数，输入是文本x，输出是文本x的长度，其中$x \in U$，Lenth(x)$\in$Z。例如：m=Lenth(p)表示模式p的长度为m，又记为$|p|=m$。
- SingleMatch(p_i,t_j)：该算法为单个字符匹配函数，输入是模式第i个字符p_i及待匹配文本的第j个字符t_j，输出值true或false。其中$p_i \in C$，$t_j \in C$，1≤i≤m，1≤j≤Lenth(t)。

1. 中文串匹配的形式化定义

- WindowMatch(p, t_j)：该算法为窗口匹配函数，输入是模式p及待匹配文本t的第j到第j+m之间的字符串，通过调用函数SingleMatch(p_i, t_j)完成窗口匹配功能，输出值true或false。
- TextMatch(p, t)：该算法为文本匹配函数，输入是模式p及待匹配文本t，通过调用窗口匹配函数WindowMatch(p, t_j)完成文本匹配功能，输出值为窗口匹配函数匹配成功的次数累计。

1. 中文串匹配的形式化定义

（3）算法性能分析

WindowMatch(p, t_j)仅对纯文本有效，对夹杂文本无效。也就是说，基于窗口的经典的字符串匹配算法无法解决夹杂文本的字符串匹配问题。

目前的基于中文关键词恶意夹杂的中文主动干扰技术可以有效地避开基于中文字符串匹配算法，使得包含这类算法的内容安全过滤/网络入侵检测手段失效。

只有改进中文字符串匹配方法才能解决恶意夹杂字符的字符串匹配问题，克服恶意的中文网络主动干扰。

2. 柔性中文字符串匹配算法

（1）算法的思想和伪代码

柔性中文字符串匹配算法采用基本的蛮力算法思想，但上一节中的形式化定义除了Lenth(x)之外均不再有效。其匹配过程可以形象地看成用一个包含中文模式p的模板沿文本t滑动，同时对文本t的每个字符位移注意模板上的字符是否与文本中的相应顺序的字符相匹配。最后统计模式匹配成功的次数，包括正常关键字个数和异常关键字个数。

2. 柔性中文字符串匹配算法

算法步骤如下：

- Step 1: 参数计算，判断是否异常退出；
- Step 2: 文本预处理，包括将文本中的拼音转化为汉字，将繁体字转化为简体字，将英文通过英汉字典转化为汉字。字典预处理，若字典为空，建立关键字典；将关键字典$p[m]$转化为拼音关键字典$q[m]$；
- Step 3: 采用滑动窗口机制，使用拼音关键字典作为模式，将文本比较一遍，分别统计文本中存在的正常模式匹配成功次数和异常模式匹配成功次数。

2. 柔性中文字符串匹配算法

算法：FlexMatch（ ）算法

输入参数：用于匹配的模式*p*，待匹配文本*v*；

输出结果：*v*中模式*p*正常个数nom，异常个数jam；

FlexMatch（*p*,*v*） {

m=Lenth（*p*）;

n=Lenth（*v*）;

if（*m*>*n*）

exit（0）;

PreProcess（*v*）;//文本预处理；

if（!*m*） SetupDict（*p*）; //若字典为空，建立关键字典；

TransDict（*p*, *q*）; //将关键字典转化为拼音关键字典；

2. 柔性中文字符串匹配算法

```
flag jam=0; //夹杂标志清0,
jam=0; // 夹杂关键字计数器清0
nom=0; //普通关键字计数器清0
for（int i=1; i<=n-m; i++） {
for （int j=1; j<=m; j++）
  { if （v[i] C） //v[i]是夹杂字符
    then { flag_jam++; i++;}
      else  if（v[i]==pp[j]） then {i++; j++;}
              else break;  }
if（j>=m） then {
     if（!flag_jam） then nom++; else jam++;  }
flag_jam=0;  }  }
```

2. 柔性中文字符串匹配算法

串匹配算法性能的影响因素很多，算法的性能表现主要取决于模式的符号分布规律和模式的长度。下面先进行中文模式(词组)均数和均长的计算，然后对FlexMatch( )算法进行分析。

- 中文模式（词组）均数与均长

根据现代汉语词典，现代汉语词组约有60000条，记为$w=60000$条。由GB2312-1980，有汉字个数$|C|_{\min}=6763$。根据GB18030-2000，有汉字个数 $|C|_{\max}=27538$。

2. 柔性中文字符串匹配算法

用 $nw = \dfrac{w}{|C|}$ 估算中文每个汉字能够组词的词组（模式）均数，计算结果如下：

$$nw_{\max} = \frac{w}{|C|_{\min}} = \frac{60000}{6763} = 8.87$$

$$nw_{\min} = \frac{w}{|C|_{\max}} = \frac{60000}{27538} = 2.18$$

$$\overline{nw} = \frac{nw_{\max} + nw_{\min}}{2} = (8.87 + 2.18)/2 = 5.53$$

2. 柔性中文字符串匹配算法

中文词的出现频率如下表：

m_i	p_i/%
单字	12.1
双字	73.6
三字	7.6
四字	6.4
多字	0.2

注：$m_i = 1, 2, \cdots, 5$表示中文词长；p_i表示中文词出现的频率百分比。

2. 柔性中文字符串匹配算法

根据上表，用 $\bar{m} = \sum_{i=1}^{5} m_i \cdot p_i$ 进行中文词组平均长度计算，中文词组（模式）均长约为 $\bar{m} = 2.078$ 字。下面的分析和计算将用到这些参数。

- 算法复杂度分析

柔性中文串匹配算法分预处理阶段和匹配阶段两部分，预处理阶段解决了异常的判断和中止处理，并使用辅助存储空间 $O(\lceil n / \bar{m} \rceil)$ 存放处理结果。

2. 柔性中文字符串匹配算法

算法核心部分是匹配阶段算法，进行以待匹配文本长度 n 为外循环、以模式长度为内循环的两重循环处理。本算法可以看作朴素的字符串匹配算法，进行模式匹配时，一旦发现一个不匹配字符或整个模式已被匹配时，朴素算法就终止对于给定位移的字符比较过程。也就是说，平均只需比较一次即离开内循环的概率是 $p_{no}=1-\frac{1}{|C|_{\min}}=\frac{6762}{6763}=0.99985$ 。因为中文平均模式长度即词组均长为 $\bar{m}=2.078$ 。如果第1个汉字匹配成功，则以词组均数倒数 $\frac{1}{nw}$ 发生第2次匹配成功，该词组匹配成功概率为 $p_{\text{Yes}}=\frac{1}{|C|}\times\frac{1}{nw}=2.67384E\text{-}5$ 。换言之。将以约0.0003的概率发生词组匹配成功，而不匹配就离开内循环的概率是 $p_{\text{No}}=1-\frac{1}{|C|_{\min}}\times\frac{1}{nw}=1-\frac{1}{6763}\times\frac{1}{5.53}\approx 0.99997$ 。如果都不匹配，则内外两重循环总共只发生 $O(n+\bar{m})$ 次匹配计算，即算法的匹配次数为待匹配文本的长度 $n+2.078$ 。这一点充分展示了本算法利用中文特征而获得的优良的匹配效率。

2. 柔性中文字符串匹配算法

柔性中文串匹配算法FlexMatch()适合在中文文本编辑及匹配等场合使用，当模式长度为m、文本长度为n时该算法的时间复杂度达到最优，为 $O(m+n)$。

当本算法处理非自然语言的小字符集文本匹配时，如英文字符集，或阿拉伯数字匹配，可能出现最坏的匹配，其时间复杂度为 $O(n*m)$。

8.2.2基于意会关键词柔性匹配的文本特征信息提取算法

英国著名心理学家和科学哲学家波兰尼在关于科学知识的研究中指出：个体的知识系统实际上有两种类型，即便于与他人沟通或交流的“言传知识”(explicit knowledge)，以及无法用言语与他人沟通的“意会知识”(tacit knowledge)。本小节引申波兰尼的研究结果，把中文关键词分类为言传关键词(explicit keyword)和意会关键词(tacit keyword)。言传关键词就是原形关键词，又称为本体关键词(ontology keyword)。意会关键词相对于言传关键词而言，是一种“只可意会不可言传”的关键词。

1. 意会关键词分类与统计方法

基于中文主动干扰的意会关键词分类定义如下：

- 定义8 中文言传关键词又称为原形关键词，它在文本中不发生任何变化，定义为0型意会（原形关键词以“蓝天白云”为例）。
- 定义9 图像意会型：关键词中有关键字或关键词用图像的方式表示，称为1型意会。本文研究对象仅限于图像中的字为印刷体，不包括手写体汉字。如1型意会关键词“蓝天白云”。

1. 意会关键词分类与统计方法

- 定义10 火星文意会型：流行于中文互联网上的一种普遍用法，融合了各种语言符号（符号、繁体字、日文、韩文、冷僻字等），用同音字、音近字、特殊符号等来替代中文汉字的“次文化用语”，由于与日常生活中使用的文字相比明显不同，文法奇异，又叫做“火星文”，本文定义为2型意会。如2型意会关键词为“1切斗4换J”，则0型意会结果是“一切都是幻觉”。火星文意会型包括夹杂繁体字关键词、夹杂同音字关键词以及拆分偏旁关键词等。

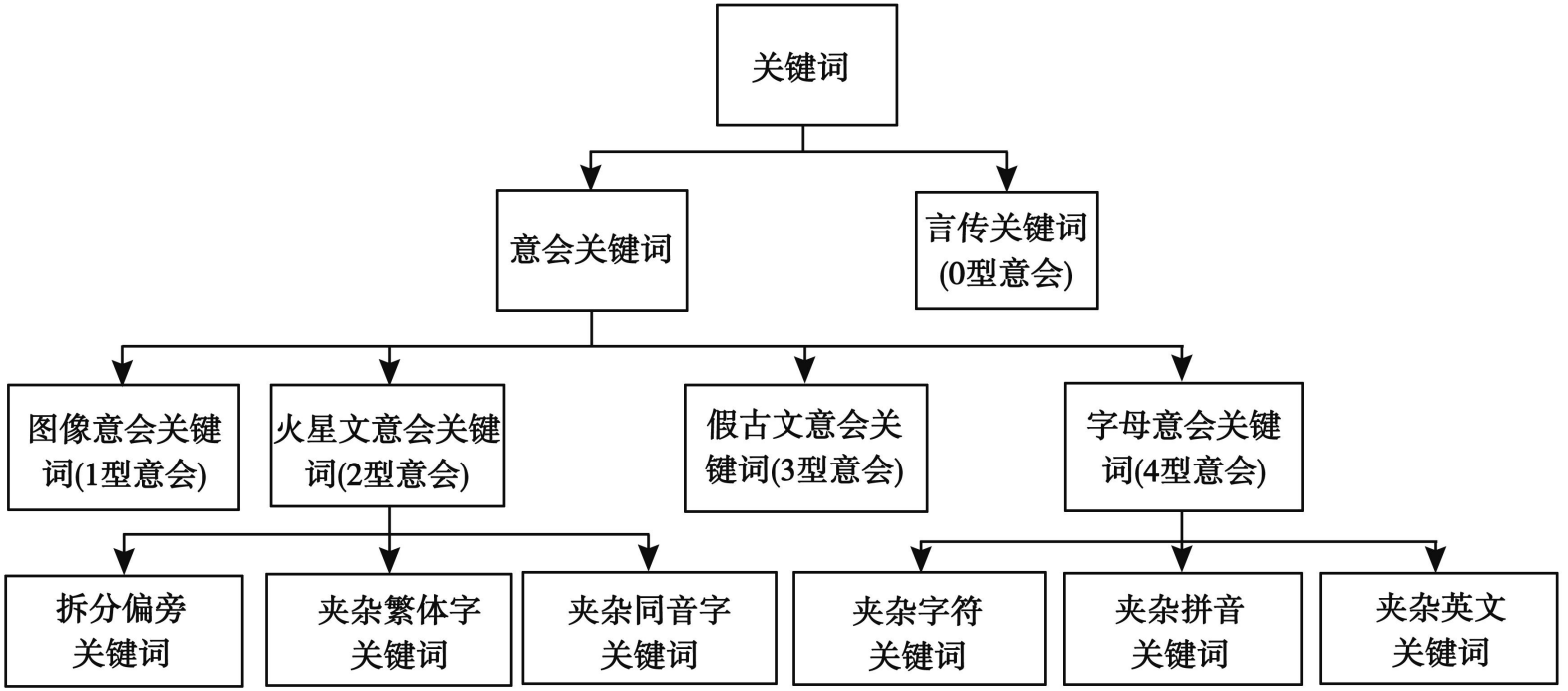
1. 意会关键词分类与统计方法

- 定义11 假古文意会型：模仿汉语古文排版，将横排文字转换成竖排文字，并自定义每一列分割符号，使认识汉字的人都能读懂内容，又有人称为“假古文”。本书定义这类变化的关键词为3型意会。如0型意会关键词为“共和的观念是：平等、自由、博爱”，3型意会结果如下图所示：

共和的观念
是：平等、
自由、博爱

1. 意会关键词分类与统计方法

- 定义12 字母意会型：在中文串中随机夹杂非汉语字符，或用汉语拼音，或用英文串代替关键词的意会方式，本书定义为4型意会。字母意会包括夹杂字符关键词、夹杂拼音关键词以及夹杂英文关键词等。如4型意会关键词为“ blue tian 白云”，其0型意会结果是“蓝天白云”。



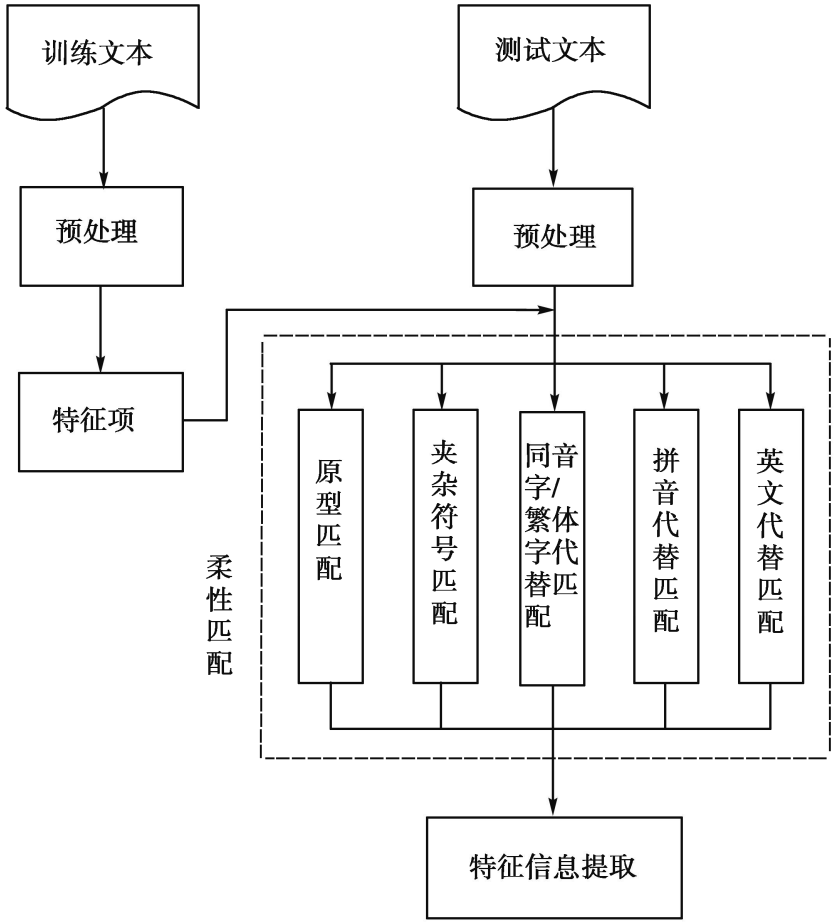
1. 意会关键词分类与统计方法

中文意会关键词统计方法：设 R_J 为查全率，P_J 为查准率，令 i 代表中文主动干扰类型（从0型意会到4型意会），T 为遭受中文主动干扰的关键词数量，T_i 表示第 i 型意会的关键词命中记录数，N 为数据库中全部关键词数（包括提取出来的和未提取出来的关键词），x 为提取出的关键词数（包括正确提取的和不正确提取出的关键词），有

$$R_J = \frac{\sum_{i=0}^{4} T_i}{N} \times 100\% \qquad P_J = \frac{\sum_{i=0}^{4} T_i}{x} \times 100\% \qquad F_J = \frac{2 \times P_J \times R_J}{P_J + R_J}$$

2. 文本特征信息提取模型

基于柔性匹配的文本特征信息提取模型：



3. 柔性匹配算法

（1）夹杂符号匹配

夹杂字符及假古文意会关键词提取算法步骤如下：

- Step1：关键词模式串在文本中向前匹配，遇到符号时直接跳过；
- Step2：若正向关键词首字匹配成功，纵向提取下一行对应列的汉字，与关键词字典进行匹配，若匹配则转Step3，否则转Step1；
- Step3：将关键词剩余部分与对应的文档进行匹配，若匹配，则转Step4，否则转Step1；
- Step4：重复前面三个步骤直至文本结束。

3. 柔性匹配算法

（2）同音字/繁体字代替匹配

步骤如下：

- Step1：根据拼音字典库，将关键词转化为其同音字/繁体字构成的关键词；
- Step2：将关键词的每种同音字/繁体字序列组合与文本进行匹配，若匹配，则转Step3，否则转Step1；
- Step3：重复前面两个步骤直至文本结束。

3. 柔性匹配算法

（3）拼音代替匹配

拼音代替意会关键词提取算法步骤如下：

- Step1：根据拼音字典库，将关键词转换为拼音；
- Step2：查找夹杂在文本中的可能为拼音的字母串；
- Step3：将此字母串与关键词的拼音形式进行第一次匹配，若匹配，则转Step4，否则转Step1；
- Step4：从文中提取出与关键词字数相对应的内容，再将其拼音串与关键词拼音串进行第二次匹配，若匹配，则转Step4，否则转Step1；
- Step5：重复前面四个步骤直至文本结束。

3. 柔性匹配算法

（4）英文代替匹配

英文代替意会关键词提取算法步骤如下：

- Step1：查找夹杂在文本中的英文；
- Step2：调用英汉数据库，将英文翻译成中文并与中文关键词采用BM算法进行初步匹配，若匹配，则转Step3，否则转Step1；
- Step3：将中文关键词和英文匹配的剩余部分与夹杂英文处左右对应的文档进行匹配，若匹配，则转Step4，否则转Step1；
- Step4：重复前面三个步骤直至文本结束。

4. 特征信息提取算法

设测试文档 d_i，特征项 t_k，利用柔性匹配技术，可以匹配出 t_k 的变形形式 $t_k^{'}$，$t_k^{'}$ 是不法分子为了逃避安全过滤，故意将进行变形的，因此 t_k 是反映文档 d_i 最重要的特征。如果不能识别出其变形形式，那么就达到了掩饰文档特征的目的；如果仅仅将识别出的变形形式归类为本体，只增加 t_k 的词频，此时也没能体现出 t_k 对文档的重要程度。所以，应该赋予有变形形式的特征项以更高的权重，综合考虑，给出权重公式如下：

$$w_{ik}=\frac{f_k^{'}(d_i)\times\log(N/N_k^{'}+0.01)}{\sqrt{\sum_{t_k\in d_i}\left[tf_k^{'}(d_i)\times\log(N/N_k^{'}+0.01)\right]^2}}\times\frac{(1+\alpha(t_k^{'}))}{\lambda}$$

4. 特征信息提取算法

$$w_{ik}=\frac{f_k^{'}(d_i)\times\log(N/N_k^{'}+0.01)}{\sqrt{\sum_{t_k\in d_i}\left[tf_k^{'}(d_i)\times\log(N/N_k^{'}+0.01)\right]^2}}\times\frac{(1+\alpha(t_k^{'}))}{\lambda}$$

式中，$f_k^{'}(d_i)$表示特征项 t_k及其变形 $t_k^{'}$在文档d_i中出现的次数（即词频）； N表示测试文本总数；$N_k^{'}$表示测试文本集中出现特征项t_k及其变形 $t_k^{'}$的文本数;$\alpha(t_k^{'})$表示特征项具有变形形式的权重因子，若特征项t_k没有变形形式，则$\alpha(t_k^{'})=0$； λ 表示比例因子，用来调节有变形形式的特征项与无变形形式的特征项间的权重。利用上式计算出每个特征项的权重，这时d就可以用向量空间来表示；根据d_i与不良文档的相似度，设定过滤阈值φ，当 $Sim(V(d_i),V(d\ ))>\varphi$ 时，则此文档 d_i为不良文档，应该过滤。

8.3 基于粗糙集与贝叶斯决策的不良网页过滤算法

不良网页过滤是网页两分类问题。本节提出了一种基于粗糙集与贝叶斯决策相结合的不良网页分类过滤算法，首先利用粗糙集理论的区分矩阵和区分函数得到网页分类决策的属性约简；然后通过贝叶斯决策理论对网页进行分类与过滤决策。仿真实验表明，该方法在不良网页分类过滤系统中开销小，过滤准确度高，因而在快速过滤不良网页的应用中具有工程应用价值。

8.3.1 引言

贝叶斯算法是以贝叶斯定理为理论基础的一种在已知先验概率与条件概率情况下得到后验概率的文本分类算法。贝叶斯分类算法原理简单，健壮性强。粗糙集理论能够获得分类所需的最小特征属性集，在不影响分类精度的条件下降低特征向量的维数，得到最简的显式表达的分类规则。采用贝叶斯和粗糙集理论相结合的方法进行不良网页过滤，可以优化分类系统的总体性能。

8.3.2 粗糙集理论

粗糙集理论是一种新的处理模糊和不确定性知识的数学工具。其主要思想就是在保持分类能力不变的前提下，通过知识约简，导出问题的决策或者分类的规则。知识约简是粗糙集理论的核心内容之一。众所周知，知识库中知识（属性）并不是同等重要的，甚至其中某些知识是冗余的。所谓知识约简就是在保持知识库分类能力不变的条件下，删除其中不相关或不重要的知识，并得到知识的最小表达。

8.3.2 粗糙集理论

设 $K=(U,P,AT,V,\rho)$为一概率知识表示系统，即U 是论域，P 是U 的子集全体构成的代数上的概率测度，$AT=\{a_1,a_2,\cdots,a_n\}$是有限个属性构成的集合，$V=V_1\times V_2\times\cdots\times V_n$，$V_i$是属性的值域，$\rho:U\to V$是信息函数，对于$U$ 中的每个对象 x，$\rho(x)$称为x 的描述，具有相同描述的对象是不可分辨的，记与 x具有相同描述的对象全体为$[x]$。设$\Omega=\{\omega_1,\omega_2,\cdots,\omega_s\}$是具有有限个特征状态的集合，每个具有状态$\omega_i$的对象是$U$的子集，常称为概念，$A=\{r_1,r_2,\cdots,r_m\}$是由 m个可能决策行为构建的集合，$P(\omega_j|[x])$表示一个对象在描述$[x]$下处于状态ω_i的概率，一般假定 $P(\omega_j|[x])$ 为已知的。令 $\lambda(r_i|\omega_j)$ 表示状态ω_i时采用决策r_i的风险损失。

8.3.3 粗糙集与贝叶斯决策的网页过滤方法

一种采用粗糙集与贝叶斯决策相结合的不良网页过滤方法，在相应的网页特征现象对应的各个网页类别下，利用粗糙集中区分矩阵和逻辑运算对网页特性现象进行知识约简，剔除判断网页类别的冗余属性，对约简后的网页特征现象进行网页类别的初步分类，建立网页类别决策初表，然后进行网页分类，通过网页归类，建立网页类别决策复表，最后通过贝叶斯决策过程来确定页面类别以及是否进行过滤。

8.3.4 算法设计

1. 风险系数

由于对网页中不良页面的确定并不一定通过存在不良信息的阈值百分之百的确定，所以算法在通过粗糙集确定不良类型页面后，根据贝叶斯准则，给予每种不良类型评定一个风险系数，用于进一步进行过滤决策，这样可以提高网页过滤的正确率和避免误过滤而带来的计算机高开销。令β为页面重要度，页面重要度分为I、II、III、IV四个等级：I为重要度小的网页（一般指普通的新闻、娱乐等网页），II为重要度中等的网页（一般指企业、公司、学校等网页），III为重要度较高的网页（一般指涉及商业秘密、网络交易等网页），IV为重要度很高的网页（一般指涉及国家机密、军事机密等网页）。

1. 风险系数

网页重要度系数表如下：

等级	I	II	III	IV
难度系数β	0.3	0.5	0.7	0.9

其中重要度系数在$0\leq\beta\leq1$，其中网页为不良或存在不良信息时未被过滤而导致的风险为e^{γ}，其中γ为网页的危害度。根据网页的信息内容，把页面危害程度也分为I、II、III、IV四个等级，危害程度为I＜II＜III＜IV。网页危害度系数如下表所示：

等级	I	II	III	IV
危害度系数γ	0.3	0.5	0.7	0.9

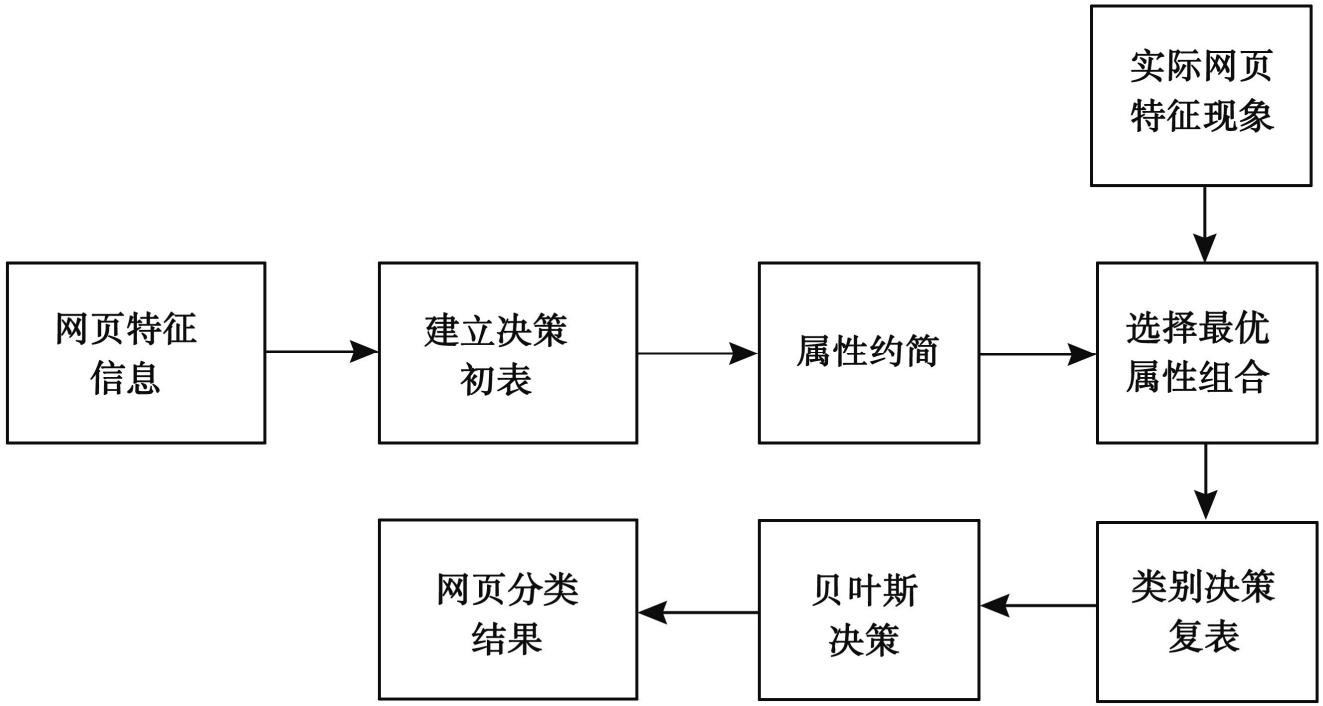
2. 过滤算法

具体步骤如下：

- Step1 收集网页特征信息，根据网页特征信息进行类别的初步分类，建立网页类别样本决策初表；
- Step2 根据粗糙集理论对网页类别样本决策初表建立相应的区分矩阵，对其进行属性约简，选择最优的属性组合，简化网页类别样本决策初表，形成网页类别决策复表；
- Step3 对于网页特征信息不在类别决策复表中的根据收集的历史特征信息以一定的先验概率确定为某种类别的网页；
- Step4 对于网页实时分类用网页类别决策复表进行决策，确定网页为某种不良类别的后验概率 $P(\omega|[x])$；
- Step5 由贝叶斯准则，确定过滤网页的风险系数 $\alpha = \frac{e^{\beta}}{e^{\beta}+e^{\gamma}}$；
- Step6 当网页为某种不良类别的后验概率为 $P(\omega|[x]) \geq \alpha$ 时，确定为不良类别并进行过滤，当 $P(\omega|[x]) < \alpha$ 时，定为非不良页面并不予过滤，最后给出决策结果。

2. 过滤算法

诊断算法流程图如下所示：



8.3.5 算例与仿真结果

1. 算法实例

假定在客户端对网页进行过滤，由算法的Step1，根据网页特征现象和网页类别与相应特征现象表，由算法Step2，建立区分矩阵C_D，用粗糙集理论对C_D进行属性约简，得到网页类别决策复表，由算法Step3，得到网页类别不确定表，由算法Step4-Step6，确定风险系数，由于考虑实际网页为普通的不良页面，风险系数中参数β定为I级，定位III级。所以

$$\alpha = \frac{\lambda_{12}}{\lambda_{21} + \lambda_{12}} = \frac{e^{0.3}}{e^{0.3} + e^{0.6}} = 0.4256 = 42.56\%$$

风险系数概率超过$\alpha = 42.56\%$ 的都可以进行网页过滤，这样可以最大程度地保护网页的安全过滤。

1. 算法实例

区分矩阵C_D如下所示：

$$\begin{bmatrix}
\Phi & a_{12} & a_{13} & a[3] & a_{15} & \Phi & a[5] & a_{18} \\
 & \Phi & a_{23} & a_{24} & a_{25} & a_{26} & a_{27} & a_{28} \\
 & & \Phi & a_{34} & a_{35} & a_{36} & \Phi & a_{38} \\
 & & & \Phi & a_{45} & a_{46} & a_{47} & a_{48} \\
 & & & & \Phi & a_{56} & a_{57} & \Phi \\
 & & & & & \Phi & a_{67} & a_{68} \\
 & & & & & & \Phi & a_{78} \\
 & & & & & & & \Phi
\end{bmatrix}$$

1. 算法实例

网页特征现象与属性值列表如下所示：

编号	特征现象	属性值
a[1]	网页内容中心思想倾向性不健康，带有不良思想。	a[1]＝0（无） a[1]＝1（有）
a[2]	宗教迷信类关键词占整个网页文本比重超过设定阈值。	a[2]＝0（不超过） a[2]＝1（超过）
a[3]	宗教迷信类语句占整个网页文本比重超过设定阈值。	a[3]＝0（不超过） a[3]＝1（超过）
a[4]	网页中Flash和视频等流媒体中出现大量裸体镜头。	a[4]＝0（无） a[4]＝1（有）
a[5]	网页中含有人物裸体图片占比重超过设定阈值。	a[5]＝0（不超过） a[5]＝1（超过）
a[6]	网页链接的IP地址、域名和URL多数在反动恶势力黑名单中。	a[6]＝0（不在） a[6]＝1（在）
a[7]	网页中背景声音中存在法轮功等宗教反动语言和声音。	a[7]＝0（无） a[7]＝1（有）
a[8]	网页文本内容潜在情感倾向性为宗教迷信类。	a[8]＝0（无） a[8]＝1（有）

1. 算法实例

网页类别及相应现象特征表如下所示：

编号	网页类别	相应特征现象
d1	正常网页	所有均正常
d2	混合不良网页	a[1],a[2],a[3],a[4], a[5],a[6],a[7],a[8]
d3	色情网页	a[4],a[5]
d4	封建迷信网页	a[1],a[2],a[3],a[8]
d5	宗教反动网页	a[1],a[2],a[3],a[6],a[7],a[8]

1. 算法实例

网页类别样本决策初表如下所示：

U	1	2	3	4	5	6	7	8
a[1]	0	1	0	0	1	1	0	1
a[2]	0	1	0	0	0	0	0	0
a[3]	0	1	0	1	1	1	0	1
a[4]	0	1	1	0	0	0	0	0
a[5]	0	1	1	0	0	1	1	1
a[6]	0	1	0	0	0	0	0	1
a[7]	0	1	0	0	0	0	0	1
a[8]	0	1	0	0	1	1	0	1
d	d1	d2	d3	d4	d5	d1	d3	d5

1. 算法实例

网页类别决策复表如下所示：

U	a[2]	a[3]	a[5]	a[8]	D
1	0	0	0	0	d1
2	1	1	1	1	d2
3	0	0	1	0	d3
4	0	1	0	0	d4
5	0	1	0	1	d5
6	0	1	1	0	d1
7	0	1	1	1	d5

1. 算法实例

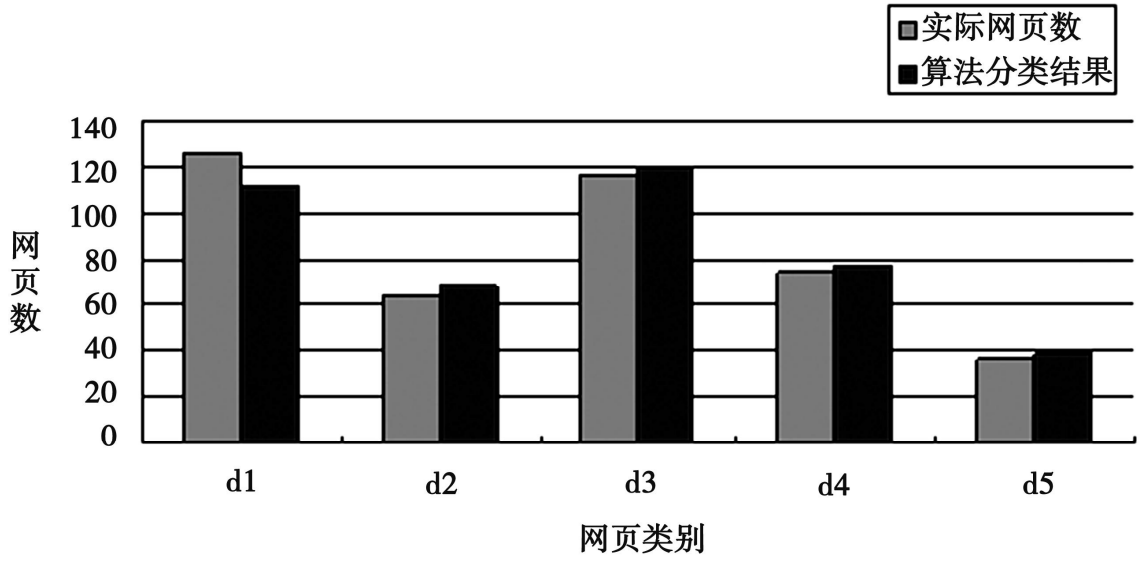
网页类别不确定表如下所示：

U	a[2]	a[3]	a[5]	a[8]	D
8	0	0	0	1	d4=80%
9	0	0	1	1	d2=60%
10	1	0	0	0	d4=60%
11	1	0	0	1	d4=90%
12	1	0	1	0	d2=60%
13	1	0	1	1	d2=80%
14	1	1	0	0	d4=70%
15	1	1	0	1	d5=40%
16	1	1	1	0	d2=70%

表中D表示网页类别的可能性，其分类概率为样本训练后得到的先验概率。

2. 仿真结果

对此算法实例进行仿真，对仿真的417组不良网页特征现象反馈数据进行网页分类，假设考虑环境人为影响和外界干扰，每组数据的可靠性为98.5%， 网页分布情况图如下所示：



图中d1表示正常页面，d2表示混合不良页面，d3表示色情页面，d4表示封建迷信页面，d5表示宗教反动页面。

2. 仿真结果

网页分布情况表如下所示：

页面类别	实际各页面数量	算法分类结果	误差
正常页面	126	112	14
混合不良网页	64	68	4
色情网页	116	120	4
封建迷信网页	74	78	4
宗教反动网页	37	39	2

2. 仿真结果

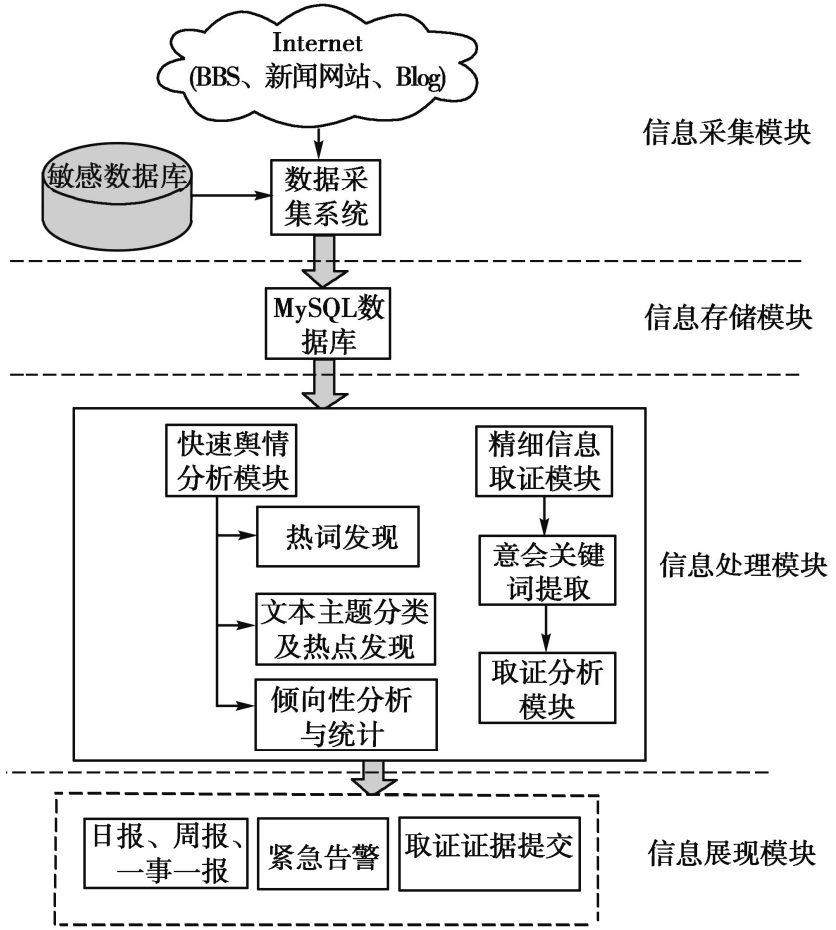
从仿真结果来看，利用本算法进行对不良网页分类过滤效果明显，并且能进一步提高过滤正确率，在对传统单用决策表进行不良网页分类过滤时，过滤正确率为88.6%（数据可靠性为98.5%）。与传统单用决策表的方法相比，本节采用的算法平均分类正确率为93.2%，过滤正确率为92.2%，与传统的算法有明显提高。这是因为网页分类过程实际上是一个搜索匹配过程。由于网页的数据庞大，这使得传统的搜索匹配过程冗余而效率低下。在本节所用的粗糙集理论对属性进行约简后再次进行匹配可以大大降低系统的冗余度，提高搜索匹配效率，也避免了大量冗余无用信息造成的误过滤，而且对于模糊类别采用贝叶斯决策可以使得过滤风险性降为最小并得到最佳分类过滤。

8.4 互联网舆情检测分析系统实例介绍

结合互联网信息传播特点，作者定制了一个互联网舆情监测分析系统，对门户、新闻、社交、博客、论坛、微博等网站中的海量信息开展实时监测和采集，借助于数据挖掘、自然语言处理等技术对所采集得到的信息进行主题检测、内容提取、自动消重、自动分类、专题聚焦，并通过统计分析自动生成时间趋势分析、话题传播分析、舆情简报、舆情专报和舆情预警，真实体现舆情动态。

8.4.1 系统概述

互联网舆情监测分析系统架构图如下所示：



8.4.1 系统概述

- **信息采集模块：**针对互联网数据（包括结构化数据与非结构化数据），进行实时寻址、采集、抽取、清洗、挖掘、处理，从而为各种信息服务系统提供精准数据支持。
- **信息存储模块：**为数据管理层，在硬件环境基础上，采用关系型数据库，建立信息管理平台数据源，包括建立舆情库、敏感词库和规则库。管理各类信息数据，采用成型的内容管理技术、知识管理技术、发布技术等通用技术，建立业务应用的基础平台。
- **信息处理模块：**通过建立舆情库，匹配敏感词和规则库实现对互联网信息（新闻、论坛等）的实时监测、采集；结合系统自身的内容管理平台，对采集的信息进行自动分类聚类、自动消重、主题检测、专题聚焦等；将采集并分析整理后的信息直接为用户或为用户辅助编辑提供信息服务，如自动形成舆情信息简报、追踪已发现的舆论焦点等。
- **信息展现模块：**将系统采集的信息和分析后的结果通过周报、日报或紧急告急等方式通过该模块展示给用户。

8.4.2 系统功能

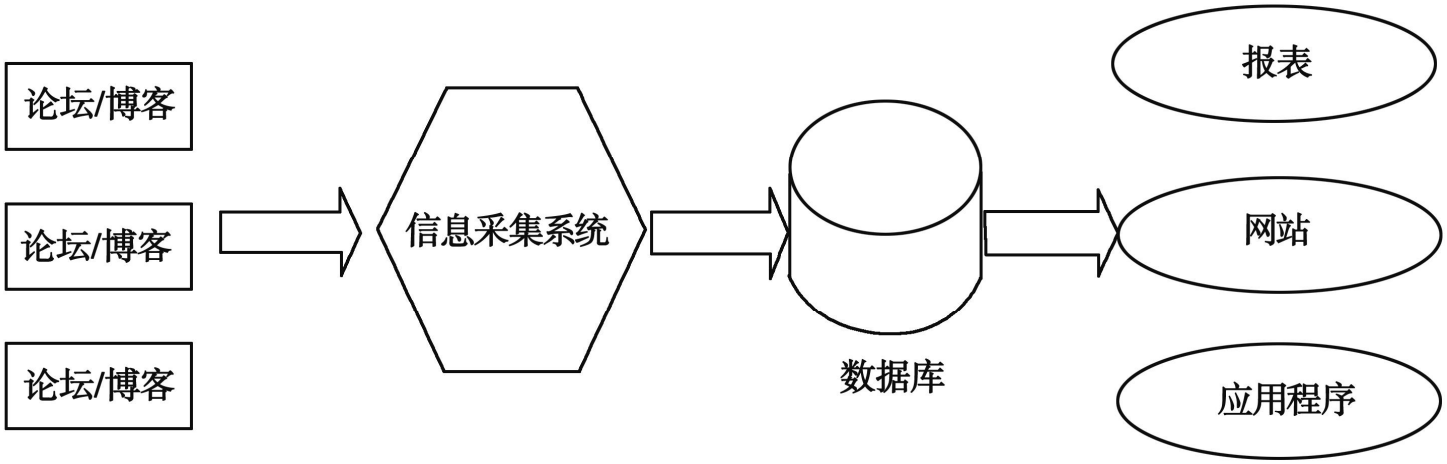
1. 舆情信息采集

本系统采用定向采集为主、全网监控为辅的方法，针对与日常业务具有密切关系的网站进行定期监控，使这类网站的任何新的信息能快速及时地被采集。

采集系统的主要功能为：根据用户自定义的任务配置，批量、精确地抽取目标论坛栏目中的主题帖与回复帖中的作者，标题，发布时间，内容，栏目等，转化为结构化的记录，保存在本地数据库中。

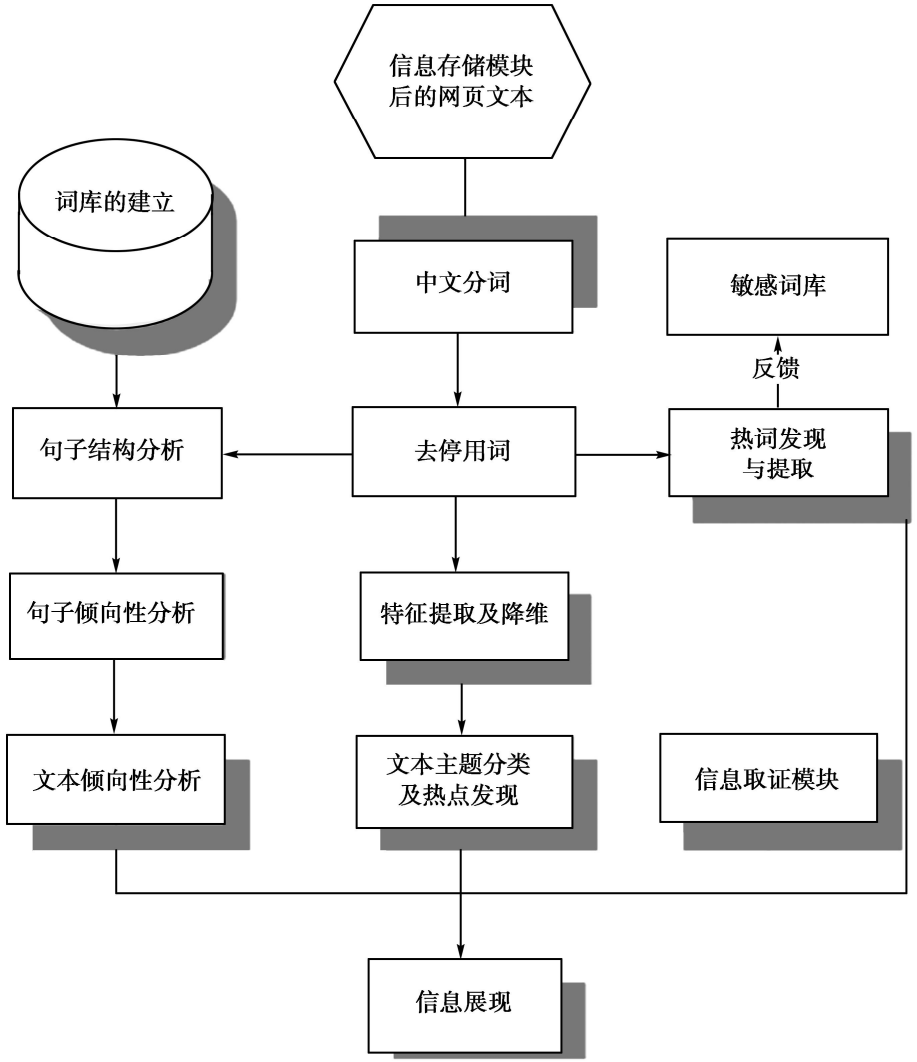
1. 舆情信息采集

功能示意图如下图所示：



2. 数据智能分析处理

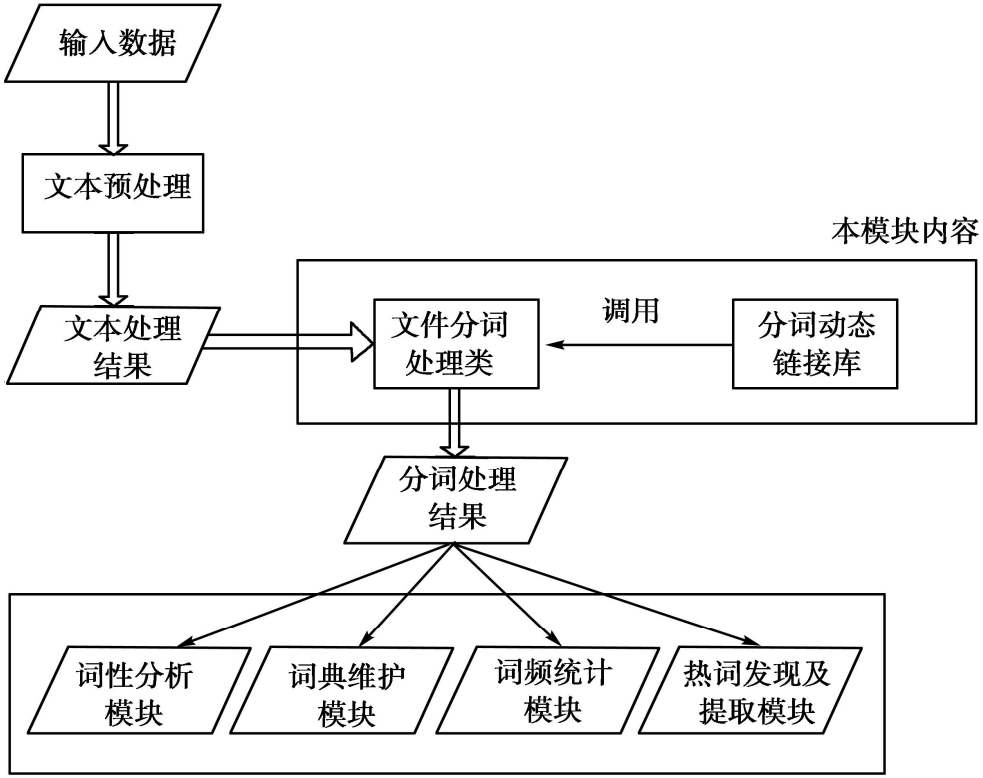
信息处理模块流程图如下图所示：



2. 数据智能分析处理

（1）中文分词模块

以基于中文分词的混合字词为索引单位，内嵌的分词系统采用以词典为基础的分词算法。系统自带一部通用的系统词典，用户可以通过建立用户词典来定义新的词汇，用户词典一般包含了某个领域的专业词汇。系统在自动分词时将同时参考缺省分词词典和用户词典中的词汇。



2. 数据智能分析处理

本模块技术指标：

- a）以PDAT大规模知识库管理技术为基础，在高速度与高精度之间折中，可管理百万级别的词典知识库，单机每秒可以查询100万词条，内存消耗不超过知识库大小的1.5倍。分词速度单机约1MB/s，分词精度98.45%。
- b）采用层叠隐马尔可夫模型，将汉语词法分析的所有环节统一到一个完整的理论框架，争取最好的总体效果。
- c）可分别处理简繁体中文；支持当前广泛承认的分词和词类标准，包括计算所有词类标注集ICTPOS3.0，北大标准、滨州大学标准、国家语委标准、台湾“中研院”、香港“城市大学”；用户可以直接自定义输出的词类标准，定义输出格式。

2. 数据智能分析处理

（2）关键词筛选及自动获取摘要

自动关键词提取是通过智能的手段为文档自动提取关键词的技术。由于本系统处理的对象主要为舆情信息包括新闻报道、评论等，我们根据词性标注结果提取出文章含有的名词、动词、名词短语，然后使用自主设计的评估函数将关键词排列，从中选取可能性较大关键词，这大大提高摘要与关键词的准确性与可读性。同时，该引擎提供静态摘要与动态摘要的功能。

实际应用系统中，在该引擎核心上可实现对文本网页等的自动提取摘要（静态摘要）与关键词，对检索结果集提供与检索条件相关的动态自动摘要，从而使检索者只需要阅读少量内容就可判断当前文档是不是所需要的文档。

2. 数据智能分析处理

（3）文本主题分类及热点发现模块

本模块的输入是分词模块对网页文本分词得到的结果，通过处理将网页划分到预先设定好的主题类别中，然后从每个类别中检测出热点并按热点的受关注度依次排列，将结果送信息展现模块，为客户全面掌握网络舆情提供有效分析依据。

实现方法：采用向量空间模型，对文档集提取特征项，对初始进行降维处理，通过计算与各类别中心点相似度的方式将文档划分到应属类别中，再在各个类别中通过质心比较策略找出热点事件及话题。

2. 数据智能分析处理

（4）自动排重与自动过滤

自动排重功能特色之处：

- a）多特征文档标识策略。
- b）智能的过滤处理。
- c）智能判断处理。
- d）动态交互特性。
- e）减小漏排率。

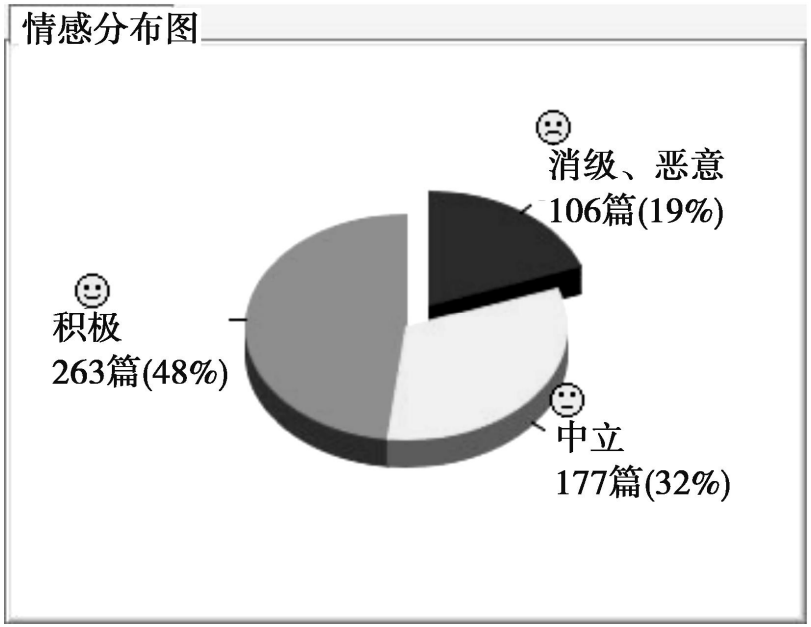
自动过滤功能特色之处：

- a）支持特征词过滤；支持特征词的布尔组合处理。
- b）支持基于事例的过滤。
- c）基于分类的过滤。
- d）过滤范围动态设置。

2. 数据智能分析处理

（5）文本倾向性分析模块

文本倾向性分析的核心是文本情感判断，是指通过机器自动学习对大量的文本集合进行情感分析，然后根据学习到的知识去对新文本的情感倾向性做预估。一般说来，将文本分为积极、消极、中立、恶意四类。情感分布图界面如下所示：



2. 数据智能分析处理

6）信息取证模块

网络舆情存在大量经过恶意处理过的非法信息需要进行计算机取证。针对非法信息取证问题，提出了意会关键词信息取证技术，该技术首先对中文意会关键词进行了定义、分类和量化，然后提出了6个意会关键词提取算法，并对提取的证据信息进行完整性处理。信息取证分析模块是本软件所特有的功能模块。

主要技术指标：

- a）基于意会关键词信息取证提取算法的提取速度为ms级；
- b）查准率和查全率分别达到了92%和95%，有效保证网页舆情监 控下的非法信息的信息取证效率；
- c）取证内容实施完整性认证处理，可作为法律证据提交有关机构或组织。

8.4.3 舆情处理结果展示

1. 关注信息

（1）网络新闻、论坛的例行报告

系统能够对重要的热点新闻进行分析和追踪，及时掌握舆情爆发点和事态。系统会根据新闻文章在各大网站和社区的传播链进行自动跟踪统计，提供不同时间段的热点新闻，并且每条热点新闻还可以查看新闻相关传播链，了解在某一时间段，该热点新闻在某些站点的传播数量，形成日报、一事一报、周报的word模板。

1. 关注信息

（2）网络舆情紧急告警报告

系统对采集的信息按照不同的级别提供紧急告警功能。

（3）舆情负面信息判断及取证

传统基于关键词匹配的信息过滤，往往导致大量正面信息也会被封杀，比如批判“法轮功”的文章也容易被过滤排除掉。系统基于统计、关键词匹配、知识库建立和句法规则等不同的技术形成信息褒贬分析，向用户提供正负面信息判断，同时获取证据。

2. 模块功能

（1）每天下午在指定的时间（例如17：00）生成并打印出当天十大热点话题及每个话题的类别、相关倾向程度；

（2）每周指定一个下午的固定时间（例如周五17：00）生成并打印出本周十大热点话题及每个话题的相关倾向程度；

（3）给定一个热词，一个时间段，能够展现该时间段内相关的网页新闻数、论坛帖子数总数，每个类别的相关的网页新闻数、论坛帖子数，相关文本倾向程度；

2. 模块功能

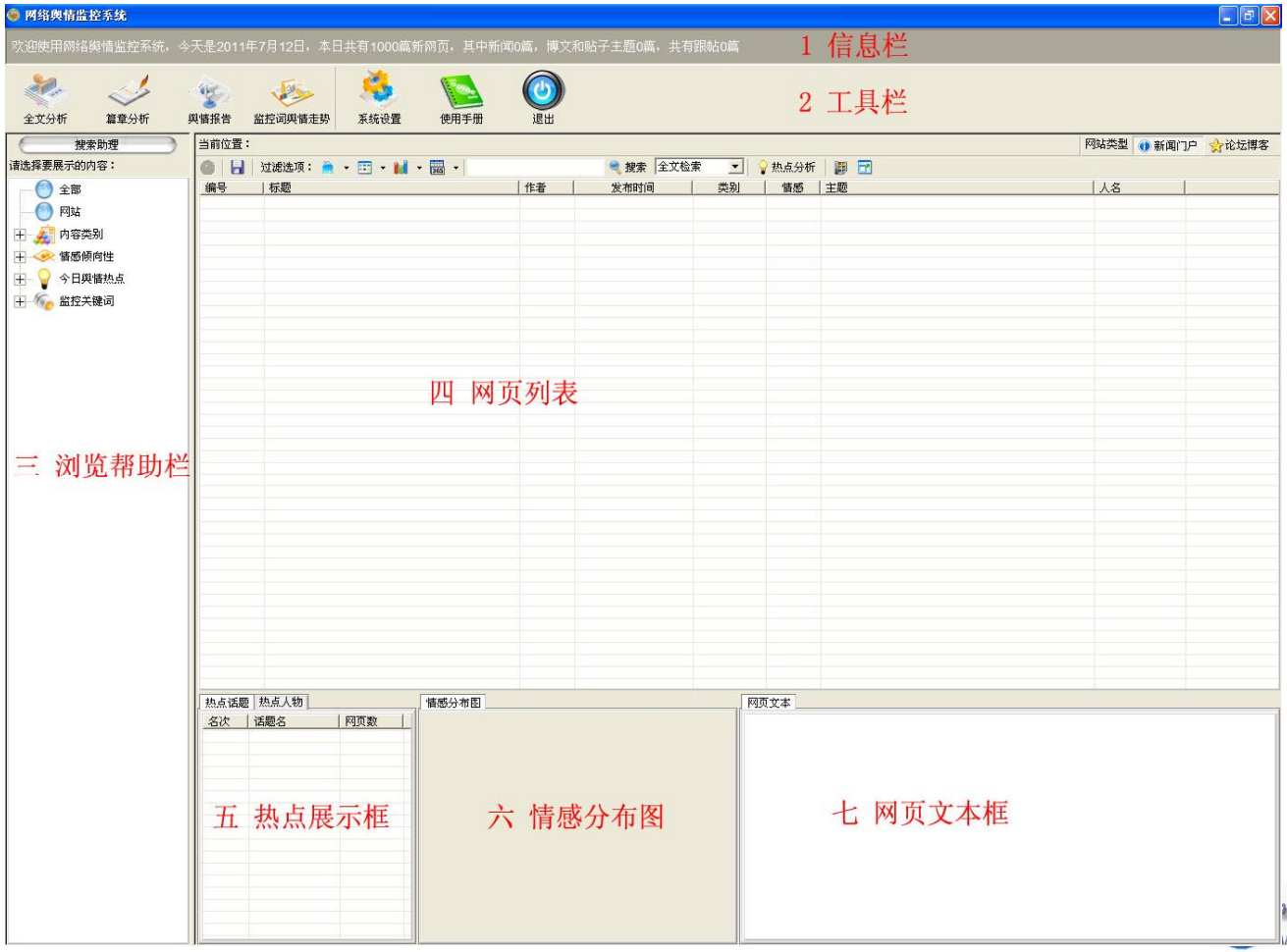
（4）设定一系列阈值，按照该阈值分成不同的级别，对24小时内网页新闻数、论坛帖子数总数超过每个阈值的话题展现出不同级别的告警；

（5）设定一个最高阈值，若某段时间内的网页新闻数和论坛帖子数总数超过了该阈值，认为该事件是高谈论对象的突发事件，即时报警并打印出该事件的相关信息，包括网页新闻数和论坛帖子数总数、时间段、类别、相关文本倾向程度；

（6）展现不良信息网页的URL、网页的日期、标题及文本内容和意会扭曲程度。

3. 信息展示形式

软件主界面如下图所示：

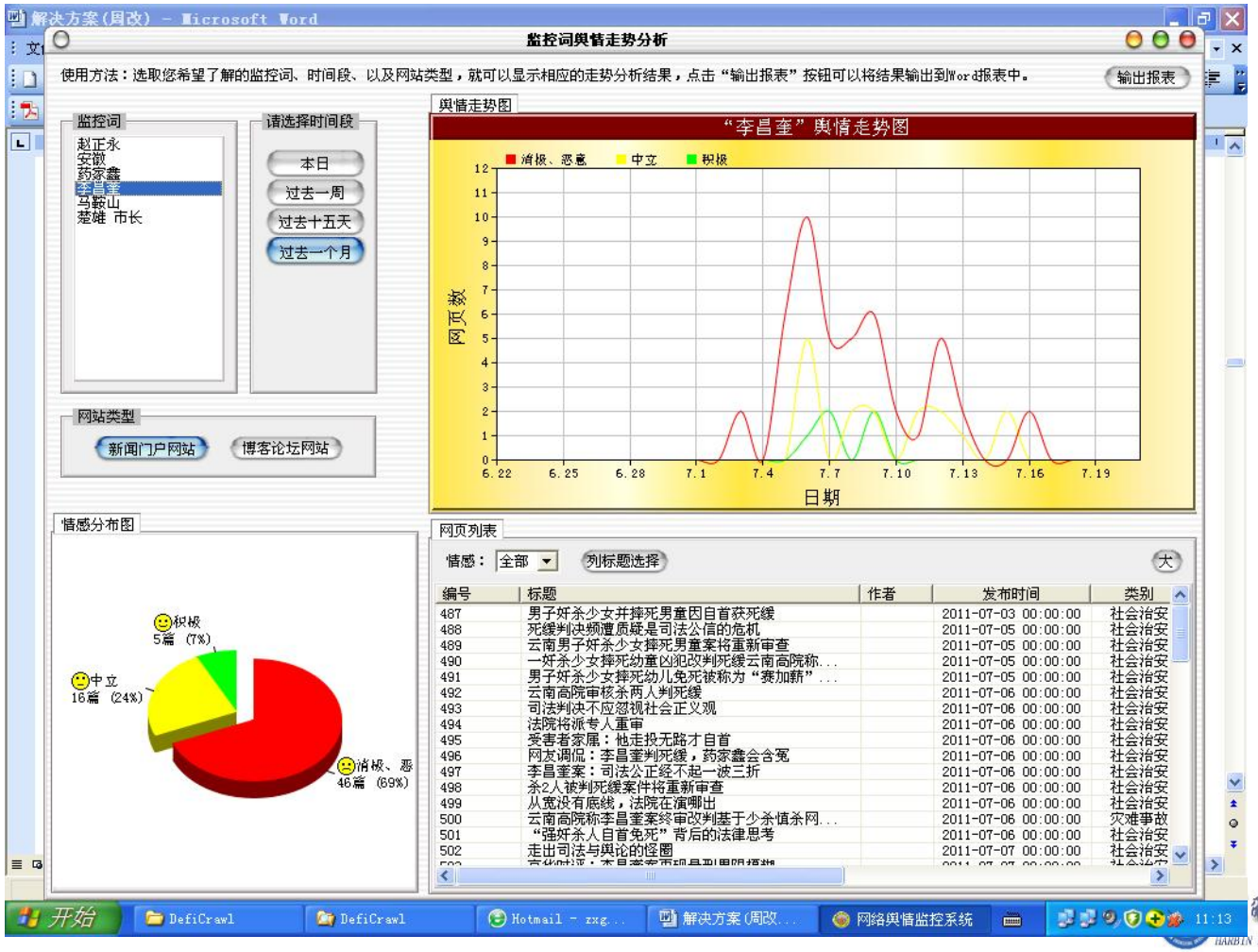


3. 信息展示形式

舆情报告界面如下图所示：

3. 信息展示形式

舆情走势分析界面如下图所示：



4. 舆情监控信息管理

- **敏感词管理：**系统支持敏感词库的管理，通过关系型数据库建立数据管理平台，可以由监管人员根据每个时期监管的对象自由定义。支持敏感词分类树管理的方式，可分组管理。
- **热点管理：**系统通过关系型数据库建立数据管理平台，支持主题管理，可以由监管人员根据自由定义主题分类，支持分组管理。
- **任务管理：**系统支持多种案件管理的方式，可以任务名、时间、处理结果等方式进行管理查询。

8.4.4 系统管理

（1）人员及权限管理

提供系统管理员相关配置选择，包括人员、日志、系统配置、公告及统计等功能。本系统提供了完善用户和权限管理机制，充分保证情报信息内容的安全性。用户分组、分类，权限分级。在视图管理环境下，可以实现对信息资讯库的访问权限的分配，对用户权力定制。通过多层次的权限控制可以达到对用户的身份甄别，对内部资源的安全保护与利用。

（2）日志管理

保存所有登录系统人员的浏览和操作历史记录，供需要参考时调用。

（3）界面定制

系统支持提供个性化的界面定制，符合各单位的办事风格，界面简洁、美观，方便用户操作，并提供直观的操作流程。

8.4.4 系统管理

（4）参数管理

系统参数主要用来设定网络采集和其他信息源、预警规则、信息分类树管理、文本挖掘配置、模板管理、信息分析服务等。

- a）信息源管理。权限范围内的员工可以选择添加新的站点、频道，或者元搜索关键词。监控和搜索参数采用标准配置文件管理，可批量导入。
- b）规则管理。本系统中采用多维矩阵式的分类结构，采用多体系分类，系统中需要分别维护各体系的分类体系的分类结构树。对信息分类树做增加，删除，修改名称等操作。
- c）文本挖掘参数配置。配置智能分析处理的相关模块参数，包括自动提取关键词、自动摘要、自动分类、自动聚类、主题检测和追踪、相似检索等参数。

8.4.4 系统管理

（5）存储管理

存储系统由四个子系统组成：元数据存储系统、索引存储系统、中心存储系统、备份存储系统。内容管理平台实现了一个分布式、多层次的体系结构，同时又具有集中管理的特性。抓取到的信息、用户的配置可分布在内部网络的多个站点，对数据库进行多重冗余备份，保证系统业务的正常运行和敏感数据和用户重要配置的安全。

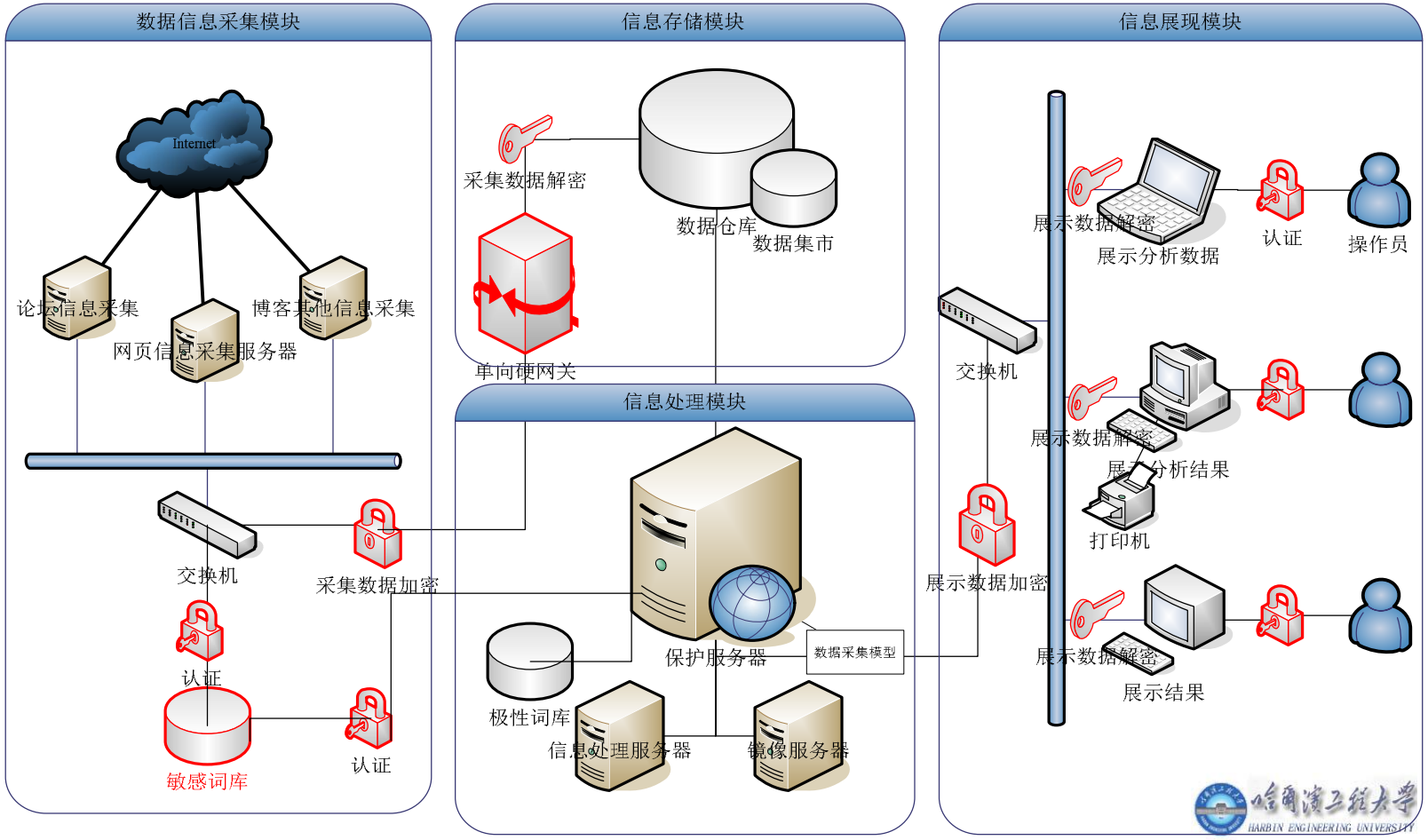
8.4.5 系统部署

1. 服务器建设内容

本系统包括信息采集服务，检索服务，分析处理服务，上述几项服务可以集中部署在一台性能较强的服务器上，也可以分散部署于多台硬件服务器，以降低主服务器的应用负载和网络带宽的占用，提高处理和查询效率。基本配置为C/S结构，服务器包括：采集服务器1台，检索服务器1台，分析服务器1台，WEB应用服务器1台，数据服务器采用1台。

2. 舆情监控系统信息防护

系统的物理结构图如下图所示：



2. 舆情监控系统信息防护

为确保系统的安全，在信息采集、信息存储、信息处理和信息展现模块中采用以下5条安全防护措施：

- （1）从Internet网上采集到的数据传入信息处理模块时，数据要求明传但又不可逆，故设置一个单向硬网关控制信息处理模块中的数据回流。
- （2）对敏感数据库（配置文件）进行加密存储在USB-key中，保证敏感数据不以明文形式出现在计算机运行环境的外部。
- （3）信息处理模块产生的结果传递给信息展现模块时采用标准加密方式传输，采用CryptoAPI方式处理。
- （4）为防止系统遭受反向工程攻击，对最终软件完成加壳处理。
- （5）操作人员采用USB-key＋用户口令方式进入系统操作，未经授权用户无法使用系统，授权操作人员配发USB-key进行身份认证。

2. 舆情监控系统信息防护

- **USB接口密码模块：**通过USB接口，安装在计算机、服务器等设备上，作为信任根，结合其他技术，构建可信计算环境。

- **USB接口密码模块配置：**舆情监控系统硬件架构采用C/S架构，其中凡是用到数据库参数配置的场合、敏感词典存放的位置、信息处理专用计算机以及信息展现平台，均需用USB-key驱动，否则，系统不提供预定服务。