文章编号: 1001-0920(2007)07-0740-05

带权约简及其在汉语词性标注自动校对中的应用

许长志[1], 闵 帆[2]
(1. 中国科学院 研究生院, 北京 100049; 2. 电子科技大学 计算机科学与工程学院, 成都 610054)

摘 要: 提出带权约简的概念,并研究了带权约简算法. 首先指出已有约简算法无法融合人类的先验知识;然后提出使用权值向量表示这类知识,用于属性重要性的计算,获得基于区分能力的带权约简算法,并分析带权约简与经典约简的关系;最后将算法应用于汉语词性标注自动校对,并讨论了权值向量的具体设置. 实验结果表明,使用所提出的算法及相应权值向量,可获得更有利于预测的约简.

关键词: 知识约简; 带权约简; 属性重要性; 汉语词性标注; 自动校对
中图分类号: TP18 **文献标识码**: A

Weighted reduction and its application in automatic correction of Chinese part-of-speech tagging

XU Chang-zhi[1], *MIN Fan*[2]
(1. Graduate School, Chinese Academy of Sciences, Beijing 100049, China; 2. School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 610054, China. Correspondent: XU Chang-zhi, E-mail: xchzh04B @mails. gucas. ac. cn)

Abstract: The concept of weighted reduct is introduced and a weighted reduction algorithm is proposed, in which the weight vector represents knowledge of human experts. The algorithm is an extension of the reduction algorithm based on discernibility, and weighted reducts are more general than traditional reducts. The algorithm is applied to automatic correction of Chinese part-of-speech tagging, and experimental results show that reducts with better prediction potential are obtained by using appropriate setting of the weight vector.

Key words: Knowledge reduction; Weighted reduction; Significance of the attribute; Chinese part-of-speech tagging; Automatic correction

1 引 言

知识约简是粗造集理论[1]的一个核心问题. 知识库中描述知识的属性并非同等重要,属性之间也可能存在依赖关系,因而导致某些属性冗余. 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、不重要或冗余的属性[1,2]. 找出所有的约简或最小约简是 NP-hard 问题[3]. 因此,人们提出了许多启发式算法(如遗传算法[4],基于信息熵[5]以及基于区分能力[6]的算法等)来寻找最小或次优约简. 上述工作都继承了粗糙集在数据分析时所具有的优点:不需要任何对数据的先验或附加信息[7]. 但在实践中,人们往往已经具备某些先验知识,而已有算法无法与之融合,以至使得所获得的结果较差.

最近,人们提出了 M- 约简的概念[8],其中 M 为用户指定包含到约简中的属性集合. 本文则假设人们对决策表中全部或部分属性的重要性已有一定了解,并使用权值向量表示这类信息,以获得基于区分能力的带权约简算法. 本文将该算法应用于汉语词性标注自动校对,实验结果表明了该算法和权值设置策略的有效性.

2 基本概念

下面介绍粗糙集理论中知识约简的相关概念,其完备定义可参见文献[1,7].

2.1 决策表与约简

三元组 $S = \langle U, C, \{d\} \rangle$ 被称为一个决策表. 其中: U 为对象的非空有限集合,称为论域; C 为条件属性的非空有限集合; d 为决策属性; $\forall a \in C \cup$

收稿日期: 2006-03-06; **修回日期**: 2006-04-30.
作者简介: 许长志(1974—),男,北京人,工程师,博士生,从事网络与信息安全技术的研究; 闵帆(1973—),男,重庆人,讲师,博士,从事粗糙集理论与方法的研究.

$\{d\}$, V_a 为属性 a 的值域,即 $a:U \to V_a$. 对于 $B \subseteq C$, $X \subseteq U$,用 $\underline{B}X$ 表示 X 的 B- 下近似集,则决策属性 d 的 B- 正区域定义为 $\mathrm{POS}_B(\{d\}) = \bigcup_{X \in U/\{d\}} \underline{B}X$. 决策表 S 的所有约简组成的集合记为 $\mathrm{RED}(S)$;决策表 S 所有约简的交集称为核,记为 $\mathrm{CORE}(S)$.

在汉语词性标注中,兼类词"累"有 3 种词性:动词(v)、形容词(a)和名形词(an). 表 1 给出了"累"在上下文环境中被标注的情况,其中属性 a_i 表示兼类词"累"的上下文环境中的词性和兼类词的词性,如 a_{-2}, a_2 与 a_0 分别表示兼类词"累"左边与右边的第 2 个位置,以及"累"本身被标注的词性. 该表为一个决策表,其中:$U = \{x_1, \cdots, x_{10}\}$, $C = \{a_{-6}, \cdots, a_{-1}, a_1, \cdots, a_6\}$, $d = a_0$.

表 1　决策表 S_1

对象	a_{-6}	a_{-5}	a_{-4}	a_{-3}	a_{-2}	a_{-1}	a_1	a_2	a_3	a_4	a_5	a_6	d
x_1	u	v	v	vn	w	d	w	c	s	d	a	w	a
x_2	nr	n	d	v	n	v	y	w	d	p	n	v	a
x_3	v	n	v	n	n	w	u	v	p	w	n	w	v
x_4	w	v	f	u	an	c	d	v	v	v	w	c	an
x_5	w	v	n	c	a	m	m	w	c	v	r	a	a
x_6	v	u	n	w	m	n	u	d	v	u	n	d	v
x_7	p	v	v	v	u	n	a	y	w	r	d	v	a
x_8	n	w	n	z	w	a	d	d	f	v	m	q	a
x_9	n	v	w	p	a	a	a	n	v	w	c	p	a
x_{10}	v	v	u	m	q	w	u	v	u	n	w	v	v

在决策表 $S = \langle U, C, \{d\} \rangle$ 中,将 $a \in C$ 的区分能力定义为 a 所区分的对象对,即

$$\mathrm{DP}(d \mid a) = \{(x_i, x_j) \in U \times U \mid d(x_i) \neq d(x_j),\ i < j,\ a(x_i) \neq a(x_j)\}, \quad (1)$$

其中 $i < j$ 保证同一个对象对不重复出现. 例如,对于决策表 S_1,有 $\mathrm{DP}(d \mid a_{-1}) = \{(x_1, x_3), (x_1, x_4), (x_1, x_6), (x_1, x_{10}), (x_2, x_3), \cdots, (x_9, x_{10})\}$. 相应地,将 $B \subseteq C$ 的区分能力定义为 B 中所有属性区分能力的并,即

$$\mathrm{DP}(d \mid B) = \bigcup_{a \in B} \mathrm{DP}(a). \quad (2)$$

例如,$\mathrm{DP}(d \mid \{a_{-1}, a_1\}) = \mathrm{DP}(d \mid a_{-1}) \cup \{(x_6, x_7)\}$. 特别地,$\mathrm{DP}(d \mid \varnothing) = \varnothing$.

有如下定理:

定理 1　对于任意 $B \subseteq C$,有

$$\mathrm{DP}(d \mid B) = \mathrm{DP}(d \mid C) \Rightarrow \mathrm{POS}_B(\{d\}) = \mathrm{POS}_C(\{d\}). \quad (3)$$

证明　根据文献[2,4]中定义及文献[2]中定理 4,$\mathrm{DP}(d \mid B) = \mathrm{DP}(d \mid C) \Rightarrow B$ 为分布协调集,而分布协调集一定是分配协调集[2],因此 $\mathrm{POS}_B(\{d\}) = \mathrm{POS}_C(\{d\})$. □

2.2　常见启发式约简算法

基本的约简算法可利用区分矩阵的计算得到[1],可以求出所有的约简,但效率很低,为指数复杂性. 由于核的计算复杂性为多项式级,人们一般先计算出核,再在其基础上利用启发式信息计算最优(最小)或次优约简. 常用启发信息包括信息熵[5]和区分能力[6]等.

3　带权约简

属性的区分能力反映了该属性与决策属性之间的依赖关系,集合 $\mathrm{DP}(d \mid a)$ 基数越大,a 的区分能力越好,其依赖关系越高. 但是,单纯的区分能力并不一定能真实地反映属性的重要性. 针对该问题,本节提出带权约简算法.

3.1　带权约简算法

对于决策表 $S = \langle U, C, \{d\} \rangle$,令 $C = \{a_1, a_2, \cdots, a_{|C|}\}$,权值向量为 $W = (w_1, w_2, \cdots, w_{|C|})$,其中 $w_i \geq 0$ 为属性 $a_i (i \in \{1, 2, \cdots, |C|\})$ 的权值.

下面给出基于区分能力的带权约简算法:

算法 1　基于区分能力的带权约简算法

输入:决策表 $S = \langle U, C, \{d\} \rangle$;

输出:S 的一个带权约简 B.

/ * 初始化,Att 表示未加入约简属性的集合 * /

Step1: $B = \varnothing$, Att $= C$;

/ * disPairs 表示 B 所能区分的对象对数 * /

Step2: disPairs $= 0$;

/ * 如果在 B 中添加 a_i,则计算新的属性集合所能区分的对象对数 * /

1) 任意 $a_i \in$Att,newDispairs$_i$ $= |\mathrm{DP}(d \mid B \cup a_i)|$;

/ * 确定最大带权属性重要性 * /

2) 任意 $a_i \in$Att,SGF$_i$ $= w_i$ * (newDispairs$_i$ - disPairs);

/ * 如果本次所涉及的所有属性的带权属性重要性均为 0,则表示 Att 中属性已经没有增加到 B 的价值 * /

3) 若对任意 $a_i \in$Att,SGF$_i$ $= 0$,则转 Step3;

/ * 将带权属性重要性最大的属性放到 B 中 * /

4) 令 SGF$_i$ 最大的属性为 a_j,Att $=$ Att $- \{a_j\}$, $B = B \cup \{a_j\}$,disPairs $=$ newDispairs$_j$;

/ * 是否还有属性可供选择 * /

5) 若 Att $= \varnothing$,则转 Step3,否则转 1);

/ * 去掉冗余的属性,后选入 B 中的属性优先

*/

Setp3: 对每个属性 $a \in B$,按其添加到 B 的逆序,计算 $POS_{B-\{a\}}(\{d\})$,如果 $POS_{B-\{a\}}(\{d\}) = POS_B(\{d\})$,则 $B = B - \{a_j\}$.

3.2 算法分析

算法 1 与基于区分能力约简算法最本质的不同在于 Step2 之 2),即属性重要性的计算过程中考虑了权值. 显然,当权值向量 W 的每个分量均相等且不为 0 时,本算法退化为基于区分能力约简算法. 从这个意义上看,本算法是对已有算法的一种推广.

算法 1 与通常使用的启发式约简算法另一个不同点是它的基础是空集,而不是核. 其原因在于核属性的权值有可能为 0,导致它不应该属于 B. 实践中可作如下改进:先计算出核,再将其中权值为 0 的属性删除,作为 B 的基础. 由于计算核属性的时间复杂度较低,在很多情况下,改进后的算法会减少执行时间. 这里为保持算法 1 的简洁性和易读性,不作相应更改.

Step3 是很必要的一步,因为它使用自顶向下的方式,最终保证 B 中没有冗余属性.

如果某个属性对应的权值为 0,根据 Step2 之 2),该属性的属性重要性一定为 0,这导致它不可能被包含进带权约简集合 B 中. 令所有权值为 0 属性组成的集合为 C',可得到如下性质:

性质 1 算法 1 的输出 B 为 S 的约简的充要条件是

$$POS_{C-C'}(\{d\}) = POS_C(\{d\}). \tag{4}$$

证明 1) C 中所有属性对应的权值为 0,最终计算出的带权属性重要性也一定为 0,因而有

$$B \cap C' = \varnothing, \tag{5}$$

$$B \subseteq C - C'. \tag{6}$$

由 Step2 之 2) 和 Step2 之 3) 可知

$$DP(d \mid B) = DP(d \mid C - C'), \tag{7}$$

根据定理 1,有

$$POS_B(\{d\}) = POS_{C-C'}(\{d\}). \tag{8}$$

而算法 1 中的 Step3 可保证 B 中没有多余的属性. 因此,如果式(4) 成立,则 B 就是约简.

2) 证明其逆否命题. 如果式(4) 不成立,则根据式(8),有

$$POS_B(\{d\}) = POS_{C-C'}(\{d\}) \neq POS_C(\{d\}), \tag{9}$$

从而 B 不是约简. □

基于性质 1,在设计算法 1 的过程中,属性选择的结束条件(见 Step2 之 4)) 是剩下属性的带权属性重要性均为 0,而不是 $DP(d \mid B) = DP(d \mid C)$ 或 $POS_B(\{d\}) = POS_C(\{d\})$.

由于权值为 0 的属性肯定不属于 B,算法 1 所针对的实际上是决策表 $S' = \langle U, C - C', \{d\}\rangle$. 由性质 1 易得如下性质:

性质 2 算法 1 的输出 B 是决策表 $S' = \langle U, C - C', \{d\}\rangle$ 的约简.

这样,决策表 S 允许权值为 0 的带权约简问题就转换为决策表 S' 不允许权值为 0 的带权约简问题. 如下推论在很多时候用起来更为简洁:

推论 1 如果 W 的分量均不为 0,那么算法 1 的输出 B 是决策表 S 的约简.

3.3 权值的作用及一般性设置策略

在带权约简的过程中,权值起到了辅助作用. 由于有了人的参与,增加了人的领域相关经验,可以避免过度依赖于训练数据. 一般情况下,权值的作用包括:

1) 删除特殊的无用属性. 在许多数据集中,有一个对象标识(ID) 属性,如表 1 中的对象名. 单从区分能力上看,该属性显得最重要,但是使用它将不具有任何的泛化能力. 通常人们在处理这种数据集时会将它预先删除,而在带权约简模型中,只需将其权值设置为 0 即可. 从形式上看,这样可以使操作一致化.

2) 降低噪音的影响. 现实数据中会不可避免地遇到噪音,完全消除噪音一般不可能,确定哪些数据是噪音也同样困难[7]. 实际上,约简本身也是一个提高泛化能力、消除噪音的过程. 在有些情况下,人们知道属性受噪音影响相对严重,这时就可以将其权值设置得相对较小,降低其被选择到约简中的可能性. 需要注意的是,不能仅因为噪音的影响,就武断地删除相应属性(将其权值设置为 0).

3) 减小有争议属性的重要性. 从已有经验的角度看,某些属性可能与决策属性不存在因果关系,但也不能完全排除它们与决策属性之间存在一定程度的关系的可能. 因此,这类属性是否应从数据集中删除存在一定的争议. 这种情况下可以将其权值设置得较小,降低其带权属性重要性.

总之,希望通过调整权值,使得约简向用户希望的结果进行,从而提高约简算法的灵活性.

现在讨论权值影响力的限制. 如前所述,通过设置权值为 0 可以删除某属性,但通过将权值设置得很高并不能保证相应属性被包含在带权约简中. 根据性质 2,只需考虑权值非 0 的情况. 假设 W 中各分量均不为 0,则有:

性质 3 给定 $a_i \in C$,可通过设置 W 而使得 $a_i \in B$ 的充要条件为

$$a_i \in \cup RED(S). \tag{10}$$

证明　1) 由于 $a_i \in \cup \mathrm{RED}(S)$,一定存在 Red $\in \mathrm{RED}(S)$, s.t. $a_i \in \mathrm{Red}$.

任选一个满足条件的 Red,以下只需证明可以设置 W 使得 $B = \mathrm{Red}$. 直接有效的办法是不属于 Red 的属性所对应的权值设置得相当小. 可采用下面的方法:

首先令

$$\varepsilon = \min_{C_1 \subseteq \mathrm{Red}} \min_{a_i \in \mathrm{Red}-C_1} | \mathrm{DP}(d \mid C_1 \cup \{a_i\}) - \mathrm{DP}(d \mid C_1) |.$$

ε 代表 Red 中所有属性的非带权属性重要性的最小值,由于 Red 是约简的,$\varepsilon > 0$. 令

$$M = \max_{C_1 \subseteq \mathrm{Red}} \max_{a_i \in C-\mathrm{Red}} | \mathrm{DP}(d \mid C_1 \cup \{a_i\}) - \mathrm{DP}(d \mid C_1) |.$$

M 代表不在 Red 中所有属性的非带权属性重要性的最大值.

Red 中所有属性的权值设置为 1, C - Red 中所有属性所对应的权值为 $\varepsilon/(M+1)$,则 C - Red 中所有属性基于 Red 或其子集的带权属性重要性,一定比 Red 中任意属性的带权属性重要性小. 因此,算法 1 的 Step2 之 5) 只可能选择 Red 中属性. 而由算法 1 的结束条件知,不在 Red 中的属性不会再被考虑.

2) 证明其逆否命题. 如果 $a_i \notin \mathrm{RED}_{\{d\}}(C)$,由于不允许 W 中任何分量为 0,根据推论 1, B 一定是 S 的约简,即 $B \in \mathrm{RED}_{\{d\}}(C)$. 因此 $a_i \notin B$. □

需要注意,性质 3 中的构造方法仅为证明其正确性,实际情况下不会进行这样的设置. 该性质说明,权值向量起到的作用是辅助性而不是决定性的. 当然,为保证某个属性被包含进最终的带权约简中,可以通过设置权值为 0,将多数重要性更高的属性删除. 但这样已经失去了带权约简本身的意义,在此不作进一步讨论.

4　应用实例

自然语言处理中,词性标注的任务就是对句子中的每个词赋予一个合适的词性. 由于词的兼类现象,一些词在不同的上下文中有不同的词性,这便为词性自动标注带来困难. 研究者已经提出多种方法[9],以提高词性标注的正确率.

从训练语料中获取兼类词的所有可能词性的真实样例得到样例库. 对每个样例,采集该兼类词被标注的上下文词性信息.

在建立词性校对决策表时,参数 k 的取值难以权衡. 如果 k 取值较小,则一些决定兼类词词性的上下文信息不被包含在校对决策表中;反之,大量无关的上下文信息又会影响校对的正确率. 利用带权约简可以有效克服该问题:首先将参数 k 设置得较大,以尽可能包含有用的上下文信息;然后对词性校对决策表中每个条件属性设置权值,使得那些可能无关的条件属性有较小的权值,而可能相关的条件属性有较高的权值;最后使用带权约简算法(算法 1)对该决策表进行约简.

在词性校对决策表 $S = \langle U, C, \{d\} \rangle$ 中,属性 a_i 的权值应与位置信息 i 有关,可设其权值函数为 $w(x)$,其中 x 是位置,满足 $w(x) = w(-x)$, $w(|x|)$ 是递减函数. 此外,希望随着 $|x|$ 的变大, $|w(|x|) - w(|x| + |\Delta x|)|$ 应该减小. 设 $y = |x|$,根据泰勒展式,有

$$|w(y) - w(y + \Delta y)| = |w'(y)\Delta y|, \quad (11)$$

则 $|w'(y)|$ 应是递减函数. 显然 $w(x) = 1/|x|$, $1/|x|^2$ 或 $e^{1/|x|}$ 都满足上述条件. 为比较不同权值的设置方式对约简的影响,下面以表 1 为例,给出不同权值下的带权约简结果,如表 2 所示.

表 2　3 种不同权值下的约简结果

方案编号	$w(x)$	B
1	$1/\lvert x \rvert$	$\{a_{-1}, a_1\}$
2	$1/\lvert x \rvert^2$	$\{a_{-1}, a_1\}$
3	$e^{1/\lvert x \rvert}$	$\{a_{-1}, a_1\}$

表 2 进一步说明了 3.3 节权值设置策略,即权值设置重点在于反映属性的重要性,而不必过于精确. 基于此,为简单计,对于 $C = \{a_{-k}, a_{-k+1}, \cdots, a_{-1}, a_1, \cdots, a_{k-1}, a_k\}$,其权值向量 W 设置为

$$\left(\frac{1}{k}, \frac{1}{k-1}, \cdots, \frac{1}{1}, \frac{1}{1}, \cdots, \frac{1}{k-1}, \frac{1}{k}\right), \quad (12)$$

即对任意的 $a_i \in C$, a_i 的权值 $w_i = 1/|i|$. 显然,当 a_i 越靠近兼类词时,其权值 w_i 越大.

本文通过实验比较带权约简与经典约简应用于词性标注自动校对中的差异. 其规则获取和应用都使用固定的算法,即分别使用粗糙集方面的权威工具 RSES 2.2[10] 中提供的覆盖算法和投票算法进行规则获取和规则使用. 实验过程如下:首先,分别采用带权约简算法(算法 1)计算带权约简和基于区分能力的约简算法(将算法 1 中各属性的权值设为 1.0)计算经典约简;其次,将两种约简后的决策表输入 RSES 2.2 中,采用 RSES 2.2 中的覆盖算法获取规则集;最后使用 RSES 2.2 中的投票算法对测试集中的样例进行标注,并将由两种约简导致的标注结果进行比较.

所谓词性标注自动校对,实质上就是对已经标注的语料中的兼类词再进行标注. 由于已经进行了标注,采集兼类词的上下文语境中的词性信息就非常容易. 实验所使用的语料库为北京大学计算语言所的 PFR 语料库. 该语料库为《人民日报》的新闻题

表 3 实验结果

兼类词	可能词性	训练集	测试集	经典约简		带权约简	
				正确率 / %	覆盖率 / %	正确率 / %	覆盖率 / %
累	{a , v , an}	10	10	40.0	50.0	66.7	90.0
临时	{d , b}	32	32	67.9	87.5	80.0	93.8
公开	{v ,a ,vn ,an ,ad}	53	52	22.7	42.3	54.8	59.6
多样	{m ,a}	9	10	50.0	40.0	75.0	80.0
统计	{v ,vn}	109	109	91.8	56.0	88.5	47.7
自动	{b ,d}	13	14	70.0	71.4	100	92.9
理想	{a ,an ,n}	35	36	48.1	75.0	81.2	44.4
向	{p ,v ,nr}	670	669	98.7	89.1	98.3	95.5
适当	{a ,ad}	28	28	96.4	100	96.4	100

材,已经进行了分词和词性标注处理.为了进行自动词性校对实验,首先将该语料库中某兼类词的样例集分为两部分,一部分用于训练样例集,另一部分作为测试样例集.实验中设置 $k = 7$,权值按式(12)设置,其实验结果如表 3 所示.

经过计算可得经典约简校对这些兼类词的平均正确率和覆盖率分别为 92.3 %和 81.1 %,而使用带权约简进行校对的平均正确率和覆盖率分别为 94.4 %和 86.0 %.因此,在词性校对中,带权约简将经典约简的校对正确率提高了 2.1 %,而覆盖率提高了 4.9 %.

5 结 语

本文提出了一个带权约简算法,并将其应用于汉语词性标注自动校对,获得了较好结果.带权约简适用于人们对某些属性的重要性、易获得性或噪音等已有部分了解的情况.它强调用户参与,突出领域知识与经验知识,这在交互式数据挖掘中很有意义.但在许多应用中,用户可能难以进行较好的权值向量设置.今后将研究权值向量的自适应算法,即用户只需表达一定的倾向,由系统进行自动调节.

参考文献(References)

[1] Pawlak Z. Rough sets [J]. Int J of Computer and Information Sciences , 1982 , 11(5) : 341-356.

[2] 张文修 , 米拒生 , 吴伟志. 不协调目标系统的知识约简[J]. 计算机学报 , 2003 , 26(1) : 12-18.
(Zhang Wen-xiu , Mi Ju-sheng , Wu Wei-zhi. Knowledge reductions in inconsistent information systems[J]. Chinese J of Computers , 2003 , 26(1) :12-18.)

[3] Wong S K M , Ziarko W. On optimal decision rules in decision tables [J]. Bulletin of Polish Academy of Sciences ,1985 , 33(11/12) : 693-696.

[4] Wroblewski J. Finding minimal reducts using genetic algorithms [C]. JCIS '95. Wrightsville Beach , 1995 : 186-189.

[5] 王国胤 , 于洪 , 杨大春. 基于条件信息熵的决策表约简[J]. 计算机学报 ,2002 , 25(7) : 1-8.
(Wang Guo-yin , Yu Hong , Yang Da-chun. Decision table reduction based on conditional information entropy [J]. Chinese J of Computers , 2002 , 25(7) : 1-8.)

[6] 徐燕 , 怀进鹏 , 王兆其. 基于区分能力大小的启发式约简算法及应用[J]. 计算机学报 , 2003 , 26(2) : 97-103.
(Xu Yan , Huai Jin-peng , Wang Zhao-qi. Reduction algorithm based on discernibility and its applications[J]. Chinese J of Computers , 2003 , 26(2) : 97-103.)

[7] Pawlak Z. Some issues on rough sets [C]. Trans on Rough Sets I. Berlin : Springer-Verlay , 2004 : 1-58.

[8] Min F , Bai Z J , He M Y , et al. The reduct problem with specified attributes [C]. Rough Sets and Soft Computing in Intelligent Agent and Web Technology , International Workshop at WI-IAT 2005. France : Compiegne University of Technology , 2005 : 36-42.

[9] 钱揖丽 , 郑家恒. 汉语语料词性标注自动校对方法的研究[J]. 中文信息学报 , 2004 ,18(2) : 30-35.
(Qian Yi-li , Zheng Jia-heng. Research on the method of automatic correction of Chinese part-of-speech tagging [J]. J of Chinese Information Processing , 2004 ,18(2) : 30-35.)

[10] Bazan J , Szczuka M , The RSES homepage[EB/OL]. (2006-11-13). http://alfa. mimuw. edu. pl/~ rses/start. html , 1994-2005.