

# 带有周期检测性的软件预防性再生策略

赵旭峰<sup>1</sup> 钱存华<sup>1</sup> NAKAGAWA Toshio<sup>2</sup>

(南京工业大学管理科学与工程学院 南京 210009)<sup>1</sup>

(Department of Marketing and Information System, Aichi Institute of Technology, Aichi 470-0392)<sup>2</sup>

**摘要** 研究了软件老化过程及软件预防性再生策略问题。引入累积损伤模型,将由 aging related bug 引起的内存的消耗看作是由于冲击造成的内存损伤量,通过周期性的检测可以得到内存被逐步消耗而不被释放的量,当内存消耗量达到预防性再生值时在下一个检测点执行软件预防性再生策略。建立了单位时间费用模型并解析得到使单位时间期望费用最低的预防性再生值,最后通过模拟进一步验证了最优再生策略的条件。

**关键词** 软件老化,软件再生,累积损伤,维护策略

**中图分类号** TP206 **文献标识码** A

## Optimal Preventive Software Rejuvenation Policy with Periodic Testing

ZHAO Xu-feng<sup>1</sup> QIAN Cun-hua<sup>1</sup> NAKAGAWA Toshio<sup>2</sup>

(College of Management Science and Engineering, Nanjing University of Technology, Nanjing 210009, China)<sup>1</sup>

(Department of Marketing and Information System, Aichi Institute of Technology, Aichi 470-0392, Japan)<sup>2</sup>

**Abstract** Software aging process and preventive software rejuvenation policy were studied in this paper. Consumption of physical memory, which would not be released, caused by aging related bug, was considered as damage suffered by shocks and could be known by periodic tests, preventive software rejuvenation was done at the next test point when consumption of physical memory reaches a preventive rejuvenation value. Expected cost rate model and optimistic preventive rejuvenation value which minimizes expected cost rate was analytically derived. Simulation experiment shows the validity of the rejuvenation conditions.

**Keywords** Software aging, Software rejuvenation, Cumulative damage, Maintenance policy

软件系统(特别是大访问量、大数据量的服务器软件)在持续运行一段时间后就会出现性能上的衰退,这种现象叫软件老化(Software Aging)<sup>[1,2]</sup>。软件老化主要是由 Heisenbug 和 Aging related bug 两类 bug 引起的<sup>[3]</sup>。软件再生(Software Rejuvenation)作为有效处理因软件老化而失效的预防性维护策略指出:当软件老化到一定程度时,暂停软件的运行,整理系统的内部状态(内存碎片清理、内存回收、刷新操作系统内核表等)并重新启动,使软件性能得到恢复。软件再生策略包括基于时间的再生策略(Time-based Rejuvenation)<sup>[1]</sup>和基于检测的再生策略(Measurement-based Rejuvenation)<sup>[2]</sup>。

近年来,一些学者对软件再生策略进行了更细致的研究: Xie<sup>[4]</sup>研究了部分与整体两种不同粒度的再生;游静等<sup>[5,6]</sup>根据不同粒度的再生策略提出了基于嵌套的软件再生策略及多级软件抗衰技术;Bobbio<sup>[7]</sup>假设软件老化是由于连续时间下离散量的积累造成的。根据这一假设,赵天海<sup>[8]</sup>讨论了基于软件老化累积损伤的再生策略,并求得了平均维护费用最低的再生时间间隔;范新媛<sup>[9]</sup>把检查点技术引入软件再生策略,

以降低系统的宕机成本。

累积损伤模型(Cumulative Damage Models)是可靠性理论中重要的模型。累积损伤过程是指系统处于随机冲击的环境中,每次冲击给系统带来一定程度的损伤,其损伤是可累加的,因而扩展为一般累积损伤过程,当累积损伤量达到预定水平  $K$  时系统发生故障<sup>[10]</sup>。Nakagawa 等<sup>[11-13]</sup>讨论了系统受随机冲击下,带有累积损伤过程的最优预防维护或替换策略。钱存华等<sup>[14-16]</sup>把累积损伤模型应用于大型数据库备份策略并给出了最优的增量及累积备份方案。

传统的基于时间的再生策略致力于求得最优的再生时间间隔,实施周期性再生。这一再生策略过于宏观,没有考虑到软件系统内部状态特征,而且频繁地执行再生将引发较高的再生成本。基于检测的再生策略从系统性能老化的评价指标(导致系统性能老化的一个主要因素或几个因素的折衷)及其阈值出发,在系统运行过程中对其进行检测,当指标达到预先设定的阈值时执行再生。在这一策略中虽然阈值可以预先给定,但是指标达到阈值是个随机事件,而且不可能做到随时检

到稿日期:2008-09-02 返修日期:2008-12-15 本文受国家自然科学基金项目(70471017,70801036),教育部人文社科规划基金(05JA630 027)资助。

赵旭峰(1983-),男,硕士研究生,主要研究方向为随机系统可靠性与维护策略及其应用, E-mail: zhao-xufeng@163.com; 钱存华(1964-),男,博士,教授,主要研究方向为随机系统可靠性与维护策略、模糊决策等; NAKAGAWA Toshio(1942-),男,博士,教授,主要研究方向为随机系统可靠性与维修策略等。

测,如何进一步做到在指标到达阈值之前进行预防性再生是值得研究的问题。因为在指标到达阈值之后软件将面临失效的高风险和再生的高成本,所以本文认为此时的软件再生策略为事后性再生策略(Corrective Software Rejuvenation)。据此,本文探讨了带有周期检测及预防性再生(Preventive Software Rejuvenation)的软件再生策略。

对物理内存的占用而不释放是软件老化的重要因素,这一老化过程是 Aging related bug 引起的。本文以内存的占用量为对象进行检测,并引入累积损伤模型,把由 Aging related bug 引起的错误看作是对内存的一次冲击,每一次冲击对内存造成一定的损伤量(本文中损伤量表示对内存的占用量),冲击的发生和损伤量的大小都是随机的,损伤量是可以累加的。基于以上假设,本文考虑以下的再生策略:

1) 采用周期检测的方式对内存的累积占用量进行检测,检测周期为  $T$  ( $T$  预先给定),检测不能减少或消除由 Aging related bug 引起的物理内存的占用量。

2) 当内存累积占用量达到阈值  $K$  时(阈值  $K$  预先给定),系统进入低效状态,此时立即进行事后性再生策略。

3) 若在  $(nT, (n+1)T]$  内冲击导致内存累积占用量达到预防性再生值  $Z$  ( $0 < Z \leq K$ ) 时,则在  $(n+1)T$  点进行预防性再生策略。在预防性再生值  $Z$  时进行软件再生的过程见图 1。

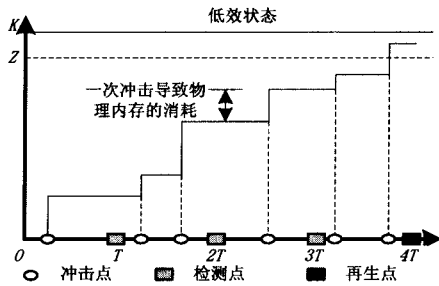


图 1 预防性再生值  $Z$  时的再生策略

## 1 模型建立

假设由 Aging related bug 引起的错误(以下用冲击替代)发生服从强度函数为  $\lambda(t)$  的非时齐 Poisson 过程,则  $R(t) = \int_0^t \lambda(t) dt$  表示区间  $(0, t]$  内平均冲击次数,在任意区间  $(t_1, t_2]$  刚好发生  $j$  次冲击的概率为:

$$H_j[R(t_2) - R(t_1)] = P_r\{R(t_2) - R(t_1) = j\} = \frac{[R(t_2) - R(t_1)]^j}{j!} e^{-[R(t_2) - R(t_1)]} \quad (j=0, 1, 2, \dots) \quad (1)$$

假设由第  $j$  次冲击产生的内存占用量(以下用损伤量替代)  $Y_j$  非负,具有有限期望且独立同分布,分布函数为  $G(x) = P_r\{Y_j \leq x\}$  ( $j=1, 2, \dots$ ),则发生  $j$  次冲击后的累积损伤量  $Z_j = \sum_{i=1}^j Y_i$  ( $Z_0 = 0$ ) 的分布函数为:

$$P_r\{Z_j \leq x\} = G^{(j)}(x) \quad (j=0, 1, 2, \dots) \quad (2)$$

其中  $G^{(0)}(x) = 1$  ( $x \geq 0$ ),  $G^{(j)}(x)$  表示  $G(x)$  的  $j$  重卷积(Stieltjes Convolution)。  $G^{(j)}(x) - G^{(j+1)}(x)$  表示累积损伤量恰好在第  $j+1$  次冲击发生时达到预定水平  $x$ 。令  $Z(t)$  表示从零时刻到时刻  $t$  的累积损伤量,则其分布函数为:

$$P_r\{Z(t) \leq x\} = \sum_{j=0}^{\infty} H_j[R(t)] G^{(j)}(x) \quad (3)$$

在时间  $(0, nT]$  内发生  $j$  次冲击,总的损伤量小于  $Z$ ,在

$(nT, (n+1)T]$  内发生  $i$  次冲击,且  $i+j$  次冲击的损伤量超过  $K$ ,软件系统进入低效状态立即进行事后性再生的概率为:

$$P_K(n, i, j) = H_j[R(nT)] H_i[R((n+1)T) - R(nT)] \times \int_0^Z [1 - G^{(i)}(K-x)] dG^{(j)}(x)$$

则进行事后性再生的概率为:

$$P_K = \sum_{n=0}^{\infty} \sum_{j=0}^{\infty} \sum_{i=0}^{\infty} P_K(n, i, j) = \sum_{n=0}^{\infty} \sum_{j=0}^{\infty} H_j[R(nT)] \sum_{i=0}^{\infty} H_i[R((n+1)T) - R(nT)] \times \int_0^Z [1 - G^{(i)}(K-x)] dG^{(j)}(x) \quad (4)$$

在时间  $(0, nT]$  内发生  $j$  次冲击,总的损伤量小于  $Z$ ,在  $(nT, (n+1)T]$  内发生  $i$  次冲击,且  $i+j$  次冲击的损伤量超过  $Z$  而小于  $K$ ,软件系统在  $(n+1)T$  点进行预防性再生的概率为:

$$P_Z(n, i, j) = H_j[R(nT)] H_i[R((n+1)T) - R(nT)] \times \int_0^Z [G^{(i)}(K-x) - G^{(i)}(Z-x)] dG^{(j)}(x)$$

则进行预防性再生的概率为:

$$P_Z = \sum_{n=0}^{\infty} \sum_{j=0}^{\infty} \sum_{i=0}^{\infty} P_Z(n, i, j) = \sum_{n=0}^{\infty} \sum_{j=0}^{\infty} H_j[R(nT)] \sum_{i=0}^{\infty} H_i[R((n+1)T) - R(nT)] \times \int_0^Z [G^{(i)}(K-x) - G^{(i)}(Z-x)] dG^{(j)}(x) \quad (5)$$

可以证明  $P_Z + P_K = 1$ 。

虽然软件再生不能消除软件本身固有的 bug,但能使软件系统恢复到初始运行状态,因此可以把软件从开始运行到再生看作一个周期,则平均周期长  $E(L)$  为:

$$\begin{aligned} & \sum_{n=0}^{\infty} (n+1)T \sum_{j=0}^{\infty} H_j[R(nT)] \sum_{i=0}^{\infty} H_i[R((n+1)T) - R(nT)] \times \int_0^Z [G^{(i)}(K-x) - G^{(i)}(Z-x)] dG^{(j)}(x) + \\ & \sum_{n=0}^{\infty} \sum_{j=0}^{\infty} H_j[R(nT)] \sum_{i=0}^{\infty} \int_0^{(n+1)T} t H_i[R(t) - R(nT)] \lambda(t) dt \times \int_0^Z [G^{(i)}(K-x) - G^{(i+1)}(K-x)] dG^{(j)}(x) \\ & = \sum_{n=0}^{\infty} \sum_{j=0}^{\infty} H_j[R(nT)] \sum_{i=0}^{\infty} \int_0^{(n+1)T} H_i[R(t) - R(nT)] dt \times \int_0^Z G^{(i)}(K-x) dG^{(j)}(x) \quad (6) \end{aligned}$$

下面引入软件再生费用:  $c_Z$  表示预防性再生费用,  $c_K$  表示事后性再生费用,  $c_T$  表示每次周期检测费用。其中  $c_K > c_Z > c_T > 0$  且  $c_K - c_Z > c_T$ 。

则一个周期内的期望费用  $E(C)$  为:

$$\begin{aligned} & \sum_{n=0}^{\infty} [(n+1)c_T + c_Z] \sum_{j=0}^{\infty} H_j[R(nT)] \sum_{i=0}^{\infty} H_i[R((n+1)T) - R(nT)] \times \int_0^Z [G^{(i)}(K-x) - G^{(i)}(Z-x)] dG^{(j)}(x) + \\ & \sum_{n=0}^{\infty} [nc_T + c_K] \sum_{j=0}^{\infty} H_j[R(nT)] \sum_{i=0}^{\infty} H_i[R((n+1)T) - R(nT)] \times \int_0^Z [1 - G^{(i)}(K-x)] dG^{(j)}(x) \\ & = c_K - (c_K - c_Z) \left\{ 1 - \sum_{n=0}^{\infty} \sum_{j=0}^{\infty} H_j[R(nT)] G^{(j)}(Z) \right\} - \\ & (c_K - c_Z - c_T) \sum_{n=0}^{\infty} \sum_{j=0}^{\infty} H_j[R(nT)] \sum_{i=0}^{\infty} H_i[R((n+1)T) - R(nT)] \times \int_0^Z [1 - G^{(i)}(K-x)] dG^{(j)}(x) \end{aligned}$$

$$R(nT)] \times \int_0^Z G^{(j)}(K-x) dG^{(j)}(x) \quad (7)$$

由式(6)及式(7)可得单位时间内的期望费用:

$$C(Z) = \frac{E(C)}{E(L)}$$

## 2 最优再生策略

由于冲击为非时齐泊松过程下讨论最优  $Z^*$  很困难,因此下面讨论冲击为时齐泊松过程下最优的预防性再生值  $Z^*$  使得单位时间期望费用  $C(Z)$  最低,即  $\lambda(t) = \lambda, R(t) = \lambda t, H_j [R(t)] \equiv H_j(t) = \frac{(\lambda t)^j}{j!} e^{-\lambda} (j=0, 1, 2, \dots)$ 。则这种情况下单位时间内的期望费用为:

$$\frac{c_K - (c_K - c_Z) \{1 - \sum_{n=0}^{\infty} \sum_{j=0}^{\infty} H_j(nT) G^{(j)}(Z)\} - (c_K - c_Z - c_T) \sum_{n=0}^{\infty} \sum_{j=0}^{\infty} H_j(nT) \sum_{i=0}^{\infty} H_i(T) \times \int_0^Z G^{(i)}(K-x) dG^{(i)}(x)}{\sum_{n=0}^{\infty} \sum_{j=0}^{\infty} H_j(nT) \sum_{i=0}^{\infty} \int_0^T H_i(t) dt \int_0^Z G^{(i)}(K-x) dG^{(i)}(x)} \quad (8)$$

由式(8)得  $C(K) = \frac{\lambda \{c_K + c_T \sum_{n=0}^{\infty} \sum_{j=0}^{\infty} H_j[(n+1)T] G^{(j)}(K)\}}{1 + M(K)}$ , 其中

$M(K) \equiv \sum_{j=1}^{\infty} G^{(j)}(K)$ 。

求式(8)关于  $Z$  的导数,并令其为 0:

$$Q_1(Z) \sum_{n=0}^{\infty} \sum_{j=0}^{\infty} H_j(nT) \sum_{i=0}^{\infty} \int_0^T H_i(t) dt \int_0^Z G^{(i)}(K-x) dG^{(i)}(x) + \sum_{n=0}^{\infty} \sum_{j=0}^{\infty} H_j(nT) \sum_{i=0}^{\infty} H_i(T) \int_0^Z G^{(i)}(K-x) dG^{(i)}(x) - U \sum_{n=0}^{\infty} \sum_{j=0}^{\infty} H_j(nT) G^{(j)}(Z) = \frac{c_Z}{c_K - c_Z - c_T} \quad (9)$$

其中

$$Q_1(Z) = \frac{U - \sum_{i=0}^{\infty} H_i(T) G^{(i)}(K-Z)}{\sum_{i=0}^{\infty} \int_0^T H_i(t) dt G^{(i)}(K-Z)}$$

$$U = 1 + \frac{c_T}{c_K - c_Z - c_T}$$

可以证明  $Q_1(Z)$  单调递增,且  $Q_1(K) = \frac{\lambda(U - e^{-\lambda T})}{1 - e^{-\lambda T}}$ 。

令式(9)等式左边为  $Q_2(Z)$ ,对  $Q_2(Z)$  求  $Z$  的导数:

$$Q_2'(Z) = Q_1'(Z) \sum_{n=0}^{\infty} \sum_{j=0}^{\infty} H_j(nT) \sum_{i=0}^{\infty} \int_0^T H_i(t) dt \int_0^Z G^{(i)}(K-x) dG^{(i)}(x)$$

由于  $G(x)$  连续且单调递增且  $Q_1(Z)$  单调递增,则  $Q_2(Z)$  单调递增,且  $Q_2(0) = 0$ ,

$$Q_2(K) = \frac{c_T}{c_K - c_Z - c_T} \left[ \frac{1 + M(K)}{1 - e^{-\lambda T}} - \sum_{n=0}^{\infty} \sum_{j=0}^{\infty} H_j(nT) G^{(j)}(K) \right] + M(K)$$

综上所述,给出以下的最优再生策略:

1) 若  $Q_2(K) > \frac{c_Z}{c_K - c_Z - c_T}$ , 即

$$c_K M(K) > c_T \left[ M(K) + \sum_{n=0}^{\infty} \sum_{j=0}^{\infty} H_j(nT) G^{(j)}(K) - \frac{1 + M(K)}{1 - e^{-\lambda T}} \right] + c_Z (1 + M(K)) \quad (10)$$

则存在唯一的  $Z^* (0 < Z^* < K)$  满足式(10)使得  $C(Z)$  最小,即在检测点发现内存占用而不释放的量达到  $Z^*$  就立即进行预防性再生,这种情况下期望费用为:

$$C(Z^*) = (c_K - c_Z - c_T) Q_1(Z^*)$$

2) 若  $Q_2(K) \leq \frac{c_Z}{c_K - c_Z - c_T}$ , 则  $Z^* = K$ , 这种情况下的期望费用为:

$$C(Z^*) = C(K)$$

## 3 模拟实验结果

给定参数  $c_K = 1000, c_T = 1, \lambda = 10, \mu = 1, K = 80, c_Z = 600, 990$ , 本文模拟出当  $Z$  在  $[40, 80]$  之间变化时对应的随机变量  $C(Z)$  的变化趋势。

由图 2、图 3 模拟实验结果可以看出最优的预防再生值  $Z^* \approx 68 < 80$ , 说明存在唯一最优的预防再生值使  $C(Z)$  最小。在这个模拟结果条件下,验证最优再生策略解析条件 1): 通过给定的参数计算出  $Q_2(K) \approx 80.18397, \frac{c_Z}{c_K - c_Z - c_T} \approx 1.5036$ , 满足  $Q_2(K) > \frac{c_Z}{c_K - c_Z - c_T}$ , 说明当满足式(10)时存在唯一的  $Z^* (0 < Z^* < K)$  使得  $C(Z)$  最小。

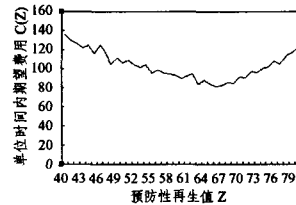


图 2  $c_Z = 600$  时的模拟实验结果 1

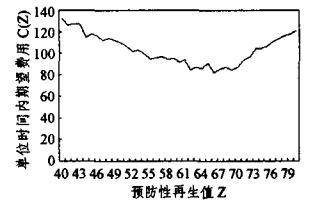


图 3  $c_Z = 600$  时的模拟实验结果 2

由图 4 模拟实验结果可以看出最优的预防再生值  $Z^*$  趋向于  $K=80$ , 说明内存占用而不释放的量达到  $K$  进行软件再生才能使  $C(Z)$  最小。进一步计算出在这模拟结果条件下,  $Q_2(K) \approx 88.15596, \frac{c_Z}{c_K - c_Z - c_T} \approx 124.27160$ , 此时不满足式(10), 因为  $Q_2(K) < \frac{c_Z}{c_K - c_Z - c_T}$ 。

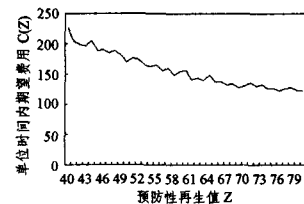


图 4  $c_Z = 990$  时的模拟实验结果

结束语 本文讨论了软件预防性再生策略,与传统的事后性再生策略不同,该策略指出当计算机资源消耗的量到达阈值之后软件进入低效状态并面临失效的高风险和再生的高成本。本文以物理内存的消耗为对象进行周期性检测,引入累积损伤过程建立期望费用率模型,并得到两种最优的再生策略使期望费用率最小,最后通过模拟的方式验证了模型最优再生策略的条件。由于可能有其它多种资源的消耗同时导致软件老化,因此本文单以物理内存为检测对象不尽合理,这

(下转第 195 页)

分,属于广义作用句的范围。

2)对统计检索结果进一步分析。按照查询关键词进行搜索,得到检索结果后,需要对其中的那些关键语句进行分析,划分其具体所属广义作用句和广义效应句范围,从中提取对应的语义块要素内容,去掉语义块对应的冗余信息,去掉相异的广义作用效应句的结果。例如:如果查询结果序列中出现“胡锦涛主席在首次出访美国期间回答了中外记者的提问。”一句,就要排除掉这个检索结果,虽然3个查询关键词都出现在句内,但是它们属于一个主语义块(“胡锦涛主席”)和一个时间辅块(“在首次出访美国期间”)内,后面的这一时间辅块的内容明显和查询要求相差甚远,需要排除,故本句明显只出现了一个查询关键词“胡锦涛主席”。在进一步分析后,应该降低它在查询结果序列中的权值。

3)对检索结果语句的语义块要素内容和查询要求分析的语义块内容进行语义块匹配计算,选取其中匹配度最大的那些结果形成二次检索结果,提供给用户。

## 5 结果分析

选取了37个语句为查询请求,对google网和百度网的前100个查询结果进行了模拟处理,输出对应的二次检索结果。应用本文给出的处理策略,对检索结果进行广义效应句和广义作用句划分以后,计算机二次检索结果得到有效提高,相对于纯统计查询检索结果不足于12%的准确率,二次检索结果的准确率提高到了87.5%,提高了近75个百分点,提升效果明显。具体结果如表2所列。

表2 正确率统计

查询句子数	用户结果在前100个检索结果中应该出现的准确率	二次挑选的结果准确率
37	11.7%	87.5%

(上接第160页)

一工作将在以后的研究过程中进一步完善。

## 参考文献

[1] Huang Y, Kintala C, Kolettis N, et al. Software Rejuvenation: Analysis, Module and Applications [C]// Proceedings of 25th Symposium on Fault Tolerant Computer Systems, Pasadena, 1995

[2] 李正, 万群丽, 许满武. 软件恢复技术研究[J]. 计算机科学, 2003, 30(8): 150-155

[3] Garg S, Van M A, Vaidyanathan K, et al. A Methodology for Detection and Estimation of Software Aging [C]// Proceedings of ISSRE, Paderborn, Germany, 1998

[4] Xie Wei, Hong Yiguang, Kishor T. Analysis of a two-level software rejuvenation policy [J]. Reliability Engineering and System Safety, 2005, 87(1): 13-22

[5] 游静, 徐建, 赵学龙, 等. 基于时间的软件恢复策略的建模与分析[J]. 计算机应用研究, 2005(11): 58-60

[6] 游静, 徐建, 李千目, 等. 计算系统多级抗衰技术[J]. 中国工程科学, 2007, 9(2): 36-43

[7] Bobbio A, Sereno M, Anglano C. Fine Grained Software Degradation Models for Optimal Software Rejuvenation Policies [J]. Performance Evaluation, 2001, 46(1): 45-62

分析这其中的原因,主要归结于优先对查询语句的广义作用句和广义效应句的判定,确定了查询目标在这两类语句中的语义块位置。分析仍然存在的错误语句,发现更多的来自于语句中的关键词部分省略情况。例如“首次出访”变成了“首访”,“出访美国”变为了“访美”等等。当然,如果需要更进一步提高检索结果的准确率,还需要进一步深入分析一次查询检索结果。彻底对检索结果语句进行语法-句法分析,从而精确提取所需的用户查询结果。

结束语 综上所述,广义作用句和广义效应句的划分,是对自然语句的一个最基本划分。HNC理论从作用和效应两极的表达侧重点不同,把自然语言的语句分为广义作用句和广义效应句,这一划分蕴含了丰富的语句知识。在海量数据检索结果的今天,如何合理利用这一划分结果,将有效提高信息检索的准确率,这种思路也为统计方法有效利用规则方法研究结果探索了一条新的道路。

需要注意的是,检索结果出现复杂句类时,广义作用句和广义效应句的判别极其困难,对二次检索的结果具有一定的影响。同时,文章上下文以及指代省略等传统的语言基本问题仍然不能简单地通过这种划分得到解决。当然这还必须以句类分析技术的根本性突破为前提条件,才能进一步提高查询结果的准确率。这些只能留在规则方法系统过程中进一步完善和提高。

## 参考文献

[1] 黄曾阳. 语言概念空间的基本定理和数学物理表达式[M]. 北京:海洋出版社,2004:25-41

[2] 黄曾阳. HNC(概念层次网络)理论[M]. 北京:清华大学出版社,1998,北京:海洋出版社,2004:175-180

[3] 游舒. HNC理论对汉语被动句的理解与处理[A]//第二届HNC与语言学研讨会论文集[C]. 北京:海洋出版社,2004:120-124

[8] 赵天海, 齐勇, 沈钧毅, 等. 基于软件老化累积损伤的再生策略[J]. 系统仿真学报, 2006, 18(2): 226-229

[9] 范新媛, 徐国治, 应忍冬, 等. 基于检查点和 Rejuvenation 的软件可靠性建模分析[J]. 系统仿真学报, 2003, 15(11): 1543-1546

[10] Cox. Renewal Theory [M]. London: Methuen, 1962

[11] Nakagawa T. On a replacement problem of a cumulative damage model [J]. Operational Research Quarterly, 1976, 27(4): 895-900

[12] Nakagawa T. A summary of discrete replacement policies [J]. European Journal of Operational Research, 1984, 17(3): 382-392

[13] Nakagawa T, Kijima M. Replacement policies for a cumulative damage model with minimal repair at failure [J]. IEEE Trans. Reliability, 1989, 38(5): 581-584

[14] Qian C H, Nakamura S, Nakagawa T. Cumulative Damage Model with Two Kinds of Shocks and Its Application to the Backup Policy [J]. Journal of the Operations Research Society of Japan, 1999, 42(4): 501-511

[15] Qian C H, Nakamura S, Nakagawa T. Optimal Backup Policies for a Database System with Incremental Backup [J]. The Transactions of the Institute of Electronics, Information and Communication Engineers of Japan, 2001, J84-A (2): 208-217

[16] Qian C H, Pan Y, Nakagawa T. Optimal policies for a database system with two backup schemes [J]. RAIRO Oper. Res., 2002, 36(2): 227-235