DOI:10.16208/j.issn1000-7024.2008.12.039

基于 Hash 结构的逆向最大匹配分词算法的改进

丁振国， 张 卓， 黎 靖
(西安电子科技大学 计算机学院，陕西 西安 710071)

摘 要：分析中文的语义，首先要对句子进行分词。中文分词是中文信息处理中最重要的预处理，分词的速度和精度直接影响信息处理的结果。对传统的分词词典和算法进行了改进，提出了基于 Hash 结构的分词词典机制，并给出了一种改进的逆向最大匹配分词算法(RMM)。该算法在重点考虑切分速度的同时兼顾了切分精度，在很大程度上消除了传统的最大匹配算法可能产生的歧义。实验结果表明，该分词算法在运行效率和结果的准确性方法有了很大的提高。

关键词：中文分词； 哈希结构； 逆向最大匹配算法； 分词词典； 消除歧义
中图法分类号：TP391 **文献标识码**：A **文章编号**：1000-7024 (2008) 12-3208-04

Improvement on reverse directional maximum matching method based on hash structure for Chinese word segmentation

DING Zhen-guo, ZHANG Zhuo, LI Jing
(College of Computer Science, Xidian University, Xi'an 710071, China)

Abstract: To analyse the Chinese semantic phrases, one must divide the sentences into words. Chinese segmentation is the most important part of Chinese information process. The speed and accuracy of segmentation influence the results of information processing. Traditional dictionary mechanisms and word segmentation methods are improved. Meanwhile, a new dictionary mechanism is provided based on hash structure, and an improved reverse directional maximum match method (RMM) is put forward. This method emphasized particularly on the speed of segmentation and the accuracy of segmentation, and it largely dispelled some ambiguities that may be produced by traditional maximum matching method. The experiment indicates that the segmentation method is improved obviously on running efficiency and veracity of the results.

Key words: Chinese segmentation; hash structure; reverse directional maximum match method; dictionary mechanism; dispel ambiguity

0 引 言

现代社会是信息社会。随着互联网上中文信息的迅速增加，如何从网上快速、准确地获取中文信息已经成为一个重要的研究课题。因此，高性能的中文信息检索系统越来越受到人们的关注。

为了避免一大堆无关信息也被检索出来，我们有必要对检索请求进行归类处理，缩小检索范围，提高检索速度和检索的准确性，即进行中文信息的处理。其中，中文分词是中文信息处理中最重要的预处理，分词的速度和精度直接影响信息处理的结果。

中文与英文不同，英文在书写时词与词之间用空格分开，而中文的书面形式却是连续的汉字串，词与词之间没有明显的标志。在中文信息处理中，大部分是基于词来进行处理的，因此必须利用分词技术进行中文词的提取。

1 分词算法简介

在所有的分词算法中，最早研究的是最小匹配算法。该算法从待比较字符串左边开始比较。先取前两个字符组成的字段与词典中的词进行比较。如果词典中有该词，则分出此词，继续从第 3 个字符开始取两个字符组成的字段进行比较；如果没有匹配到，则取前 3 个字符串组成的字段进行比较。依此类推，直到取到的字符串长度等于预先设定的阈值。如果还没有匹配成功，则从待处理字串的第 2 个字符开始比较。如此循环。这种方法的优点是速度快，但是准确率却不是很高，因此该方法基本上已经不被采用。

第 2 种分词算法是基于字符串的最大匹配算法。它分为正向和逆向两种。正向最大匹配的基本思想是：假设词典中最大词条所含的汉字个数为n个，取待处理字符串的前n个字作为匹配字段，查找分词词典。若词典中含有该词，则匹配成

收稿日期：2007-07-04 E-mail：zhangzhuo9826@sina.com
基金项目：国家 863 高技术研究发展计划基金项目(2004AA1Z2520)；军队网络互联与信息安全策略研究基金项目(2006QB1069)。
作者简介：丁振国(1959－)，男，陕西三原人，博士，教授，研究方向为计算机网络与信息处理技术； 张卓，女，硕士研究生，研究方向为计算机网络与信息处理技术； 黎靖，男，硕士研究生，研究方向为计算机网络与信息处理技术。

功，分出该词，然后从被比较字符串的n+1处开始再取n个字组成的字段重新在词典中匹配；如果没有匹配成功，则将这n个字组成的字段的最后一位剔除，用剩下的$n-1$个字组成的字段在词典中进行匹配。如此进行下去，直到切分成功为止。目前，正向最大匹配方法作为一种基本的方法已被肯定下来，但是由于错误比较大，一般不单独使用。如字符串“处理机器发生的故障”，在正向最大匹配方法中会出现歧义切分。该字符串被分为：处理机、发生、故障，但是使用逆向最大匹配就能得到有效的切分。从理论上来说，根据汉语语句中中心语一般偏后的特点，逆向匹配的精确度高于正向匹配。

逆向最大匹配 RMM(reverse directional maximum matching method)的分词原理和过程与正向最大匹配相似，区别在于前者从文章或者句子的末尾开始切分，若不成功则减去最前面的一个字。对于字符串“处理机器发生的故障”，采用 RMM 切分的结果为：故障、发生、机器、处理[1]。

中文分词还有其它分词方法，如神经网络法、联想－回溯法等，可以查阅相关文献，本文不再一一叙述。

2 词典设计

传统分词词典存在两方面的问题：一个是采用纯文本方式构建词表，数据没有经过有效的组织，内部查找的计算复杂度为$O(n)$(n为词表中词条数)；另一个是最大匹配长度的确定，中文词的字数个数以 2 为主，但普遍存在着不定长的现象，如表 1 所示[2]。

表 1 词条分布情况

词典字数	1	2	3	4	5	6	7
词条数	2606	33 527	3639	3622	83	36	3

此时 MAXL 的长度很难确定。如果定义为词典的最大汉字数，则每次分词都有若干次没有意义的循环，效率不高，浪费时间；如果 MAXL 的长度定得比较短，则一些分词匹配不到，引起分词错误。而且我们的词典可以不断丰富，词典中的最长字数也是动态在变，因此需要对词典进行改进。改进后的词典机制如图 1 所示[3]。

由图 1 可知，本词典结构由 3 部分组成：

(1) 首字 Hash 索引：首字 Hash 函数根据汉字的国标区位码给出，通过一次哈希运算即可直接定位汉字在首字 Hash 表中的序号。

首字 Hash 索引的每个单元包括 3 项内容：①关键词(2 bytes)：词的第一个汉字A；②首字下最大词长(1 bit)：标示以汉字A为首字的词的最大词长；③第一项指针(4 bytes)：指向以汉字A为首字的所有词语的起始位置。

(2)词索引：词索引的每个单元包括两项内容：①所有词长(1 bit)：标示以汉字A为首字的词的所有词长；②词典正文指针(4 bytes)：指向以汉字A为首字的相应词长的词在词典正文中的起始位置。

(3)词典正文：以词为单位的有序表。通过首字 Hash 索引表、词索引表和词典正文的配合，很容易实现指定词在词典正文中的快速查找。

3 改进的 RMM 分词算法设计

从前面对最大匹配算法(MM 算法和 RMM 算法)的介绍，我们可以看出：MM 算法和 RMM 算法都遵循“长词优先”的原则，即认为对同一个句子来说，切分所得的词数量少时是最佳切分结果。这一原则虽然会引发一些切分错误，但在大多数情况下是合理的。然而仔细分析，我们会发现一些问题：首先，两个算法都是以分词词典中最大词条所含的汉字个数n为匹配的初始最大词长，根据表 1 可得知，这样的做法会造成很多无用的循环匹配，浪费时间；其次，“长词优先”这一原则都是在局部范围内进行的，即每次最大匹配的范围都是最先i个或最后i个字符，这样并没有充分体现“长词优先”的原则。例如以下两个句子[4]：

句子 1：“当中华人民共和国成立的时候”

句子 2：“当他看到小孩子时”

如果用 MM 法进行分词，第 2 个句子的结果是：“当/他/看到/小孩子/时”，切分是正确的。但第 1 个句子的结果却是：“当中/华人/民/共和国/成立/的/时候”，显然切分是错误的。

如果用 RMM 法进行分词，第 1 个句子的结果是：“当/中华人民共和国/成立/的/时候”，切分是正确的。但第 2 个句子的结果是：“当/他/看到/小孩/子时”，显然切分时错误的。

可以看到，以上两种分词方法都在一定情况下产生了歧义切分。这里歧义产生的原因是没有充分体现“长词优先”的原则。“中华人民共和国”和“小孩子”都是句子里最长的词，但是在某些情况下被切分开来。

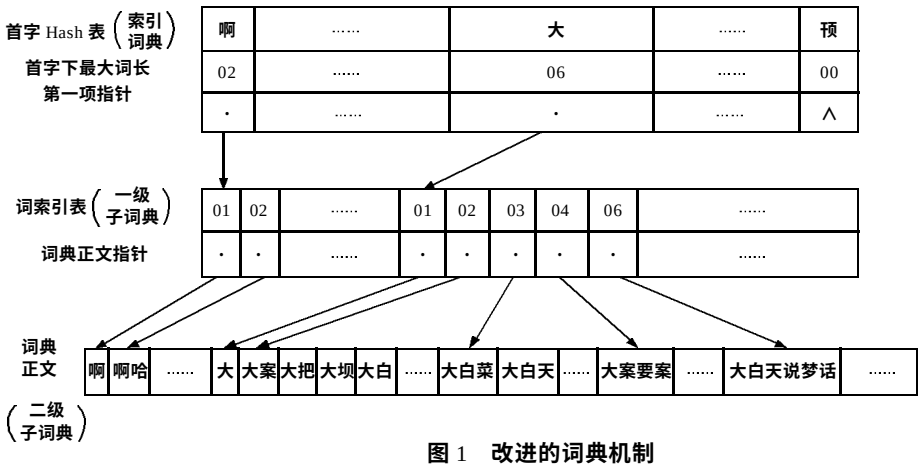


图 1 改进的词典机制

为了更合理的确定初始最大词长以及避免上述一些歧义切分，必须在整个句子的范围内实现“长词优先”的原则。为此，我们提出了一种改进的RMM算法。其算法流程如图2所示。

由图2可知，本算法有两方面的改进：

(1) 初始最大词长的选取。取出待分词字串中的每个汉字，在分词词典中查找以每个汉字为首字的词的最大词长，选出其中最大者与待分词字串的长度进行比较后，确定出最合适的初始最大词长。

(2)匹配过程的改进。从待分词字串的最后一个字n开始截取长度为i的字串，令它同词表中的词条依次匹配。如果在词表中找不到一个词条能同当前字串匹配，就从第$n-1$个字开始截取长度为i的字串重复以上过程。如果还找不到，则依次从第$n-2$，$n-3$…个字开始截取长度为i的字串进行匹配。如果在某一次匹配中查到词表中确有这样一个i字词，匹配成功，就把这个字串作为一个词从待分词字串中切分出去，把原句中位于这个字串左右两边的部分视为两个新的句子，递归调用这一过程。如果所有的匹配都不成功，说明句子中没有长度为i的词，则开始寻找长度为$i-1$的词。重复这个过程直到整个句子被切分。此过程在整个句子的范围内寻找最长词，充分体现了“长词优先”的原则。

4 系统的实现

系统采用Java语言开发，打包过后，实现一次编译，随处运行。图3为系统的处理流程图，主要分为3个阶段，分别是

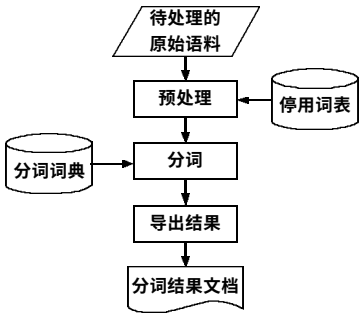


图3 分词系统处理流程

预处理阶段、分词阶段和结果处理阶段。

4.1 预处理阶段

在这一阶段中，通过对待处理的原始语料中的每个汉字进行扫描，达到两方面的效果。

(1)过滤掉没有意义的或者用户不需要的词或词组，比如“的”、“和”、“在”、“是”等以及人称的单复数称谓词，如“你”、“我”、“它”、“你们”等。同时，许多阿拉伯数字也不能存在于汉语的词汇中间，因此也将其过滤掉。对于这类词汇或标志不需要把它们切分出来，我们把这些词收集起来放在停用词词表中。停用词[5]的特点是在所有文档中出现的频率都很高，但对文档内容主题的贡献却很小。因此，在分词前剔除这些没有意义的词汇或标志是非常有必要的，这使得在分词过程中比较次数平均减少$(step-1)*step/2*I$次(I为待处理文本中停词符号的个数，$step$为比较步长)。

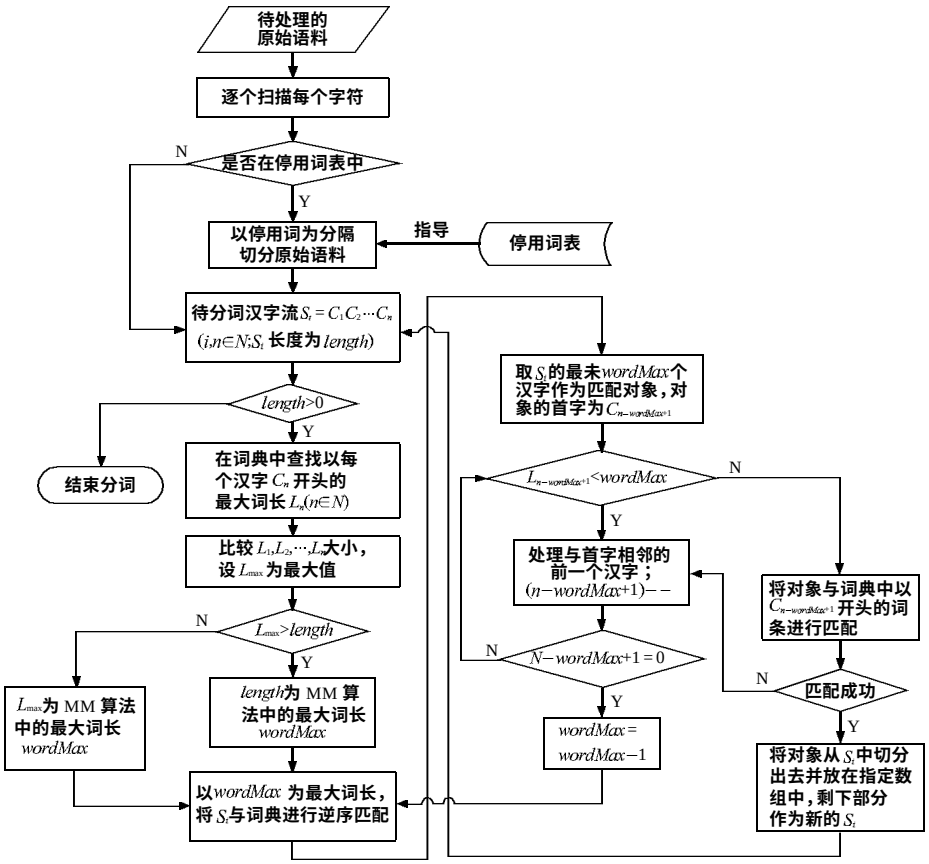


图2 改进的RMM分词算法流程

(2)扫描每个汉字的同时查询分词词典中的索引词典(如图1所示)。通过查询“首字下最大词长”这一项,获得以每个汉字为首字的词的最大词长,保存这些信息,在实际分词时就可以直接用到这些结果。

4.2 分词阶段

4.2.1 汉字编码体系[6]

汉字在计算机内部是以内码的形式进行存储的。汉字内码是汉字在汉字信息处理系统中最基本的表达形式。它与汉字交换码、汉字区位码有一定的对应关系。由于自定义编码顺序的特殊性,因此可以通过计算偏移量的方法来定义该汉字在编码表中任意的位置。国标 GB2312 汉字编码表共收录了 6763 个汉字,汉字在编码表中的偏移量计算公式如下

$$offset=(c1-0Xb0)*94+(c2-0xA1)$$

式中:$offset$——某汉字在汉字编码表中的位置,$c1$、$c2$——汉字的内部码($c1$代表内部码的高字节,$c2$代表内部码的低字节)。

本文中的首字 Hash 索引是采用汉字内码来计算的。

4.2.2 分词词典的查找算法

假设待分词字串为$S=C_1C_2\cdots C_n$ $(n\in N)$,L为根据本文中的改进 RMM 分词算法计算出的初始最大词长,则查找算法的步骤为:

(1)取S的最末L个字串,即$C_{n-L+1}\cdots C_n$。如果$n-L+1>0$,则转(2);否则转(7)。

(2)查询分词词典中以C_{n-L+1}为首字的词的最大词长M_{n-L+1}。如果$M_{n-L+1}<L$,说明$C_{n-L+1}\cdots C_n$这个字串不是一个词,转(3);否则,转(4)。

(3)$n=n-1$,转(1)。

(4)根据C_{n-L+1}的内码计算出C_{n-L+1}在编码表中的偏移量,从而确定其在首字 Hash 表中的位置。根据第一项指针确定其在词索引表中的位置,从此位置开始依次向后查找是否与L匹配。若无匹配者,则说明$C_{n-L+1}\cdots C_n$这个字串不是一个词,转(3);否则,转(5)。

(5)根据相应位置下的词典正文指针确定其在词典正文中的位置。从此位置开始依次向后查找是否与字串$C_{n-L+1}\cdots C_n$匹配的词。如果查找成功,则转(6);否则,转(3)。

(6)将字串$C_{n-L+1}\cdots C_n$从待分词字串 S 中切分出去,将剩下部分按照字串$C_{n-L+1}\cdots C_n$的左边和右边分成两个新的待分词字串,分别取出待分词字串,记为S,转(1)。

(7)$L=L-1$,转(1)。

例:查询S=“你简直是大白天说梦话”。根据本文中改进的 RMM 分词算法得到初始最大词长$L=6$,$n=10$。

(1)取出S的最末 6 个字串,即“大白天说梦话”。由于$n-L+1=10-6+1>0$,转(2)。

(2)查询分词词典中以“大”为首字的词的最大词长M_5,得$M_5=6$。由于$M_5=L$,转(4)。

(3)(4)根据“大”的内码确定其在首字Hash表中的位置,根据第一项指针确定其在词索引表中的位置,转(5)。

(5)从此位置开始找到词长为 6 的词典正文指针,由此指针定位到词典正文中以“大”为首字、词长为 6 的词。即“大白天说梦话”,转(6)。

(6)将字串“大白天说梦话”从S中切分出去,令S为剩下字串“你简直是”,转(1)。

4.3 结果处理阶段

这一阶段主要是将分词后的结果进行冗余处理,并给结果标注相应的权值。权值反映了该词的重要程度,即与用户查询意图的接近程度,最后将结果信息保存在分词结果文档中。

5 实验及结果分析

运行效率和结果的准确度,通常是衡量分词算法优劣的非常重要的标准。在其它条件相同的情况下,如果运行效率和准确度越高,则说明这种分词算法的性能越好,分词速度越快。

5.1 运行效率分析

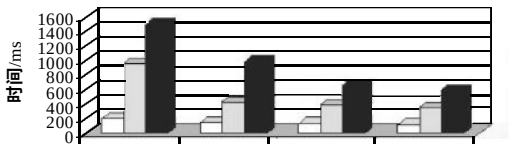
根据本文设计的分词词典机制,在查找某一个词时,首先根据“首字 Hash 表”查出该字在词表中的起始位置,然后根据该字对应的“第一项指针”查到该字在“词索引表”中的起始位置,接着根据此时最大词长,从起始位置开始根据对应词长的“词典正文指针”查到以该字为首字的词在“词典正文”中位置,从而定位到所要查找的词。若其中的任何一个步骤出现“查找失败”,即没有找到匹配者,则停止接下去的步骤,进行下一轮的查找。这样可以快速定位要查询的词在词表中的位置,减少查词的时间,提高切分速度,从而提高分词效率。

5.2 准确性分析

从本文第 4 部分的例子可以看出,改进前的最大匹配分词算法(MM 法和 RMM 法)都在一定情况下产生了歧义切分,从而导致了切分结果的不正确。这里歧义产生的原因是改进前的最大匹配法都是在局部范围内进行最大匹配,每次最大匹配的范围都是最先i个或最后i个字符,这并没能充分体现“长词优先”的原则。在例子中,“中华人民共和国”和“小孩子”都是句子里最长的词,但是在某些情况下被切分开来。而改进之后的算法,其核心思想就是在整个句子范围内寻找最长词。如对于句子 1:“当中华人民共和国成立的时候”,首先被匹配成功的字串是这个句子中最长的分词单位“中华人民共和国”。同样,对于句子 2 中的“小孩子”是句子中最长的分词单位,首先被切分出来。因此,本文所提出的算法充分体现了“长词优先”的原则,成功地处理了改进前最大匹配算法不能正确切分的句子。

5.3 实验结果分析

通过与已有的 3 种典型词典机制[7](基于整词二分的分词词典机制、基于 TRIE 索引树的分词词典机制[8]和基于逐字二分的分词词典机制)的实验分析比较,结果如图 4 所示。



	整词二分	逐字二分	TRIE 索引树	本文的 Hash 索引
□ 实验一	202	150	141	120
▒ 实验二	968	441	401	360
■ 实验三	1498	988	652	590

图 4 各词典机制分词时间分布

(下转第 3265 页)

参考文献:

[1] Liu M L.分布式计算原理与应用[M].顾铁成,王亚丽,叶保留,译.北京:清华大学出版社,2004.

[2] 佟强.基于 Mobile Agent 的分布式空间数据服务模型[J].计算机工程,2004,30(1):68-70.

[3] Jou Ahyh-Hong,Kao Shang-Juh.Agent-based infrastructure and an application to internet information gathering[J].Knowledge and Information System,2002,4(1):80-95.

[4] Moreau L.Distributed directory service and message router for mobile agents[J].Science of Computer Programming,2001,39:249-272.

[5] Guan Jihong,Zhou Shuigeng,Bian Fuling,et al.Building distribu-ted web GIS: A mobile-agent based approach [J]. Wuhan University Journal of Natural Sciences,2001,6(1-2) 474-481.

[6] 陈剑,周意青,刘振华.Java Applet 的安全性及应用[J].计算机应用研究,2001,18(11):69-70.

[7] 张冠群,陶先平,李新.移动 Agent 的迁移机制研究[J].计算机科学,2001(9):69-72.

[8] 王红,曾调,林守勋.基于 Java 的强迁移[J].小型微型计算机系统,2002(2):250-252.

(上接第 3211 页)

6 结束语

本文提出的基于 Hash 结构的分词词典机制是一种简洁、高效的词典组织模式,并且本文提出的改进的最大匹配算法是一种分词速度快、准确度高的分词算法,最大程度地满足了中文自动分词系统的现实需要。

参考文献:

[1] 张李义,李亚子.基于反序词典的中文逆向最大匹配分词系统设计[J].现代图书情报技术,2006(8):42-45.

[2] 李振星,徐泽平,唐卫清,等.全二分最大匹配快速分词算法[J].计算机工程与应用,2002,38(11):106-109.

[3] 严蔚敏, 吴伟民. 数据结构 [M]. 北京: 清华大学出版社, 2002: 251-263.

[4] 郭辉,苏中义,王文,等.一种改进的 MM 分词算法[J].微型电脑应用,2002,18(1):13-15.

[5] 熊文新,宋柔.信息检索用户查询语句的停用词过滤[J].计算机工程,2007,33(6):195-197.

[6] 张培颖,李村合.一种中文分词词典新机制——四字哈希机制[J].微型电脑应用,2006,22(10):35-36.

[7] 李庆虎,陈玉健,孙家广.一种中文分词词典新机制-双字哈希机制[J].中文信息学报,2002,17(4):13-18.

[8] 翟伟斌,周振柳,蒋卓明,等.汉语分词词典设计[J].计算机工程与应用,2007,43(1):1-2.

(上接第 3214 页)

4 结束语

设备故障诊断技术已发展为一门独立的跨学科的综合信息处理技术,是目前研究领域的热点。将 CAN 总线技术应用于设备故障诊断系统中,在硬件上可增加系统的可靠性、便于容错设计、易于系统扩充或改型、减少走线;在软件上可使通信更加灵活、实时性更好、纠错能力更强。此外与人工智能技术相结合,提高了诊断的准确率,使系统能准确反映和分析设备的运行状态[8]。

一旦控制系统出现故障,就能根据故障诊断信息以及处理方法迅速排除故障,并使系统能自动启动总线,恢复正常运行,从而有效减少设备故障停机率,是今后进一步研究的方向。

参考文献:

[1] 赵春明,乔旭彤,马宁,等.基于 CAN 的电动汽车分布式控制系统的故障诊断研究[J].车辆与动力技术,2005,22(11):41-45.

[2] 郭强,刘文仲.基于 CAN 总线的级联漏电火灾报警系统[J].计算机工程与应用,2007,43(12):246-248.

[3] 柳谦,刘震宇,龙剑飞.一种可靠的 CAN 总线多点通信设计方法[J].计算机工程与设计,2005,26(5):1323-1326.

[4] 彭文峰,吕海宝,王继东.PLC 及 CAN 总线技术在发动机故障诊断中的应用[J].机电一体化,2005,24(2):81-83.

[5] 孙立辉.基于CAN总线的多机表决式故障诊断系统的设计方法[J].仪表技术与传感,2004,33(12):38-40.

[6] 李恩,蔡丽,梁自泽,等.一种适用于煤矿安全监控系统的 CAN 总线应用层通讯协议[J].计算机应用,2006,26(9):2178-2181.

[7] 江岳春,腾召胜,张向程,等.基于 CAN 总线的电力远程监管系统主站软件设计[J].计算机工程与设计,2004,25(7):1197-1199.

[8] 孙树文,杨建武,张慧慧,等.CAN 总线在车辆分布式控制系统中的应用[J].微计算机信息,2007,3(2):45-47.