一种用于关键词检索的快速字符串精确匹配算法①

赵俊杰 (安徽财经大学 成人教育学院 安徽 蚌埠 233061)

摘 要: 在对 BF 算法、KMP 算法、BMH 算法、SUNDAY 算法和 ZZL 算法等几种常见算法分析的基础上，提出一种用于关键词检索的快速字符串精确匹配算法，并通过实验得出基本结论。最后指出模糊匹配和精确匹配的主要区别并对全文总结。

关键字: 精确匹配；关键词检索；VC 算法；模糊匹配

A Calculate Way of Rapid String Precision Used for Keyword Index Matches

ZHAO Jun-Jie

(Adult Education College, Anhui University of Finance & Economics, Bengbu 233061, China)

Abstract: Based on the analysis of BF calculate way, KMP calculate way, BMH calculate way, SUNDAY calculate way and ZZL calculate way, this paper puts forward a rapid string precision calculate way used for a keyword index, and gets basic conclusion through an experiment. Finally it points out that the faintness matches the main differentiation that matches with precision and tally up to the full text.

Keywords: precision matches; keyword inspectional; VC calculate way; misty match

1 引言

字符串匹配在计算机领域有着广泛的应用，它可用于数据处理、数据压缩、文本编辑、信息检索等多方面。如何改进字符串匹配算法，提高查询速度和效率，是目前研究的重要领域之一。

字符串匹配问题常见的分类包括精确串匹配、随机串匹配和近似串匹配。所谓精确字符串匹配问题，是在文本 S 中找到所有与查询 P 精确匹配的子串[1]。字符串精确匹配要求匹配严格准确，其实现算法主要有 BF 算法、KMP 算法、BMH 算法、SUNDAY 算法和 ZZL 算法等。本文在对这几种常见算法分析的基础上，提出一种用于关键词检索的快速字符串精确匹配算法。

2 常见算法分析

2.1 BF 算法

BF(Brute Force)算法[2]是效率最低的算法。其核心思想是：S是文本串，P是模式串。首先S[1]和P[1]比较，若相等，则再比较S[2]和P[2]，一直到P[M]为止；若S[1]和P[1]不等，则P 向右移动一个字符的位置，再依次进行比较。如果存在t，1≤t≤N，且S [t+1..t+M]= P[1..M],则匹配成功；否则失败。该算法最坏情况下要进行M*(N-M+1)次比较，时间复杂度为O(M*N)。

2.2 KMP 算法

KMP(Knuth-Morris-Pratt)算法[3]是 D.E. Knuth、J.H.Morris 和 V.R.Pratt 等人于 1977 年提出来的。其核心思想是：在匹配失败时，正文不需要回溯，而是利用已经得到的“部分匹配”结果将模式串右移尽可能远的距离，继续进行比较。这里要强调的是，模式串不一定向右移动一个字符的位置，右移也不一定必须从模式串起点处重新试匹配，即模式串一次可以右移多个字符的位置，右移后可以从模式串起点后的某处开始试匹配。KMP 算法的时间复杂度是O(m+n)，最坏情况下时间复杂度为 O(m*n)。空间复杂度是 O(m)，对 BF 算法进行了很大的改进。虽然，

① 基金项目:教育部社科研究基金青年项目(07JC870006);安徽省哲学社会规划项目(AHSKF07-08D03); 安徽财经大学教研重点项目(ACJYZD200914)
收稿时间:2009-05-21

KMP 算法有着它自身的局限性，但是，在现代科学的众多领域中的应用仍然普遍。

2.3 BMH 算法

1980 年,Horspool 提出了一种 BMH 算法[4]，它首先比较文本指针所指字符和模式的最后一个字符，如相等再比较其余 m－1 个字符。无论文本中哪个字符造成了匹配失败,都将由文本中和模式最后一个位置对应的字符来启发模式向右的滑动。即文本自位置 I 起与模式的从右至左的匹配检查中,一旦发现不匹配，就将I重新赋值为 End+d[T[End]](End是一个中间变量,记录了文本中每次从右至左匹配的起始位置)。

2.4 Quick Search 算法

Quick Search 算法[5](亦称 Sunday 算法)是 Daniel M.Sunday 于 1990 年提出的一种快速搜索算法。该算法的核心思想是：在匹配过程中，模式串并不被要求一定要按从左向右进行比较还是从右向左进行比较,它在发现不匹配时，算法能跳过尽可能多的字符以进行下一步的匹配，从而提高了匹配效率。按照这种字串移动算法，然后按照字符从右向左的次序匹配。如果完全匹配了，则匹配成功；否则，再进行下一轮的移动，直到正文 T 的最右端结束。该算法最坏情况下的时间复杂度为 O(N*M)。对于短模式串的匹配问题，该算法执行速度较快。

2.5 ZZL 算法

ZZL 算法是纪福泉等人提出的一种可作特殊用途的字符串匹配算法。现有的字符串匹配算法不论是按照模式串从左至右还是从右至左的顺序匹配，都是直接进行比较，而 ZZL 算法的核心思想是：首先在主串 S 中查找模式串 T 的首字母，每找到一个则将它的位置存储，然后依次提取这些位置，从这些位置开始继续匹配模式串 T。对于频繁使用的要匹配的主串和模式串来说，由于预先保存了模式串在主串中的所有存储位置，所以匹配速度会非常快。

3　用于关键词检索的快速字符串精确匹配算法

3.1 基本思想

在关键词检索时，应先判断这两个词是否长度相等，如果不相等就不用比较具体字符了；如果相等，首先对模式串的首尾字符比较，如果匹配再比较中间的字符。这样做的目的是有很多单词前面部分相同，例如词根相同,但后面不同,这样很多时候等匹配到最后才发现不能完全匹配,而浪费了大量时间。而在开始时就比较首尾字符，如果都相等再比较其余字符，这样匹配成功的概率就大些。这种算法的设计思想来源于测量工具游标卡尺(Vernier Caliper)的用法，所以可以简称为 VC 算法。

3.2 具体步骤

(1) 将文档分词并对每个单词统计串长，并存储在一个二维数组中(单词与串长)；

(2) 输入关键词并计算其串长；

(3) 读取第一个单词，与关键词进行以下比较：

a. 比较串长，若相等则进入下一步，否则比较下一个单词；

b. 比较第一个与最后一个字符是否相等，若相等则进入下一步，否则比较下一个单词；

c. 指针内移，依次比较中间的字符；

d. 若完全匹配则记录其位置，否则不记录；

(4) 接着读取下一个单词进行比较，直至整个数组匹配结束；

(5) 输出匹配结果。

3.3 实验与基本结论

3.3.1 实验结果

本人选取了 4 个不同长度的关键词和五篇短文，短文均在 500~600 个字符左右，大约 100~120 个单词。由于经过串长比较的初步筛选和首尾字符比较的二次筛选，最后真正进行全词比较的单词很少，不到 5%。在与 KMP 等算法相比较时，发现匹配次数明显减少。表 1 是四个关键词和其中一篇短文的匹配结果，表 2 是 VC 算法与 KMP 算法比较结果。其中短文有 661 个字符(计空格)，112 个单词。短文内容如下：

The first thing to find the similar texts to the assigned ones from thousands of texts is to judge the classification, that is, categorization. After that, we make a further calculation to find all the texts which are similar to the content of the assigned texts. Due to the similarity between text similarity calculation and text categorization, we can make use of the results of the assigned text categorization, classifycation and vector value of text feature. Then we find the texts which exceed the threshold by

making a further calculation about the similarity to other texts. In that way we can find the ones similar to the content of the assigned texts.

表 1 四个关键词和其中一篇短文的匹配结果

	串 长	串长相等的	首尾字符相等的	匹配单词数
judge	5	17	1	1
feature	7	9	1	1
similarity	10	3	3	3
categorization	14	5	3	3

表 2 VC 算法与 KMP 算法比较字符总数对比 单位：个

	KMP 算法	VC 算法
judge	656	149
feature	637	135
similarity	574	142
categorization	592	158

3.3.2 基本结论

大多数字符串匹配算法没有对原文事先分词，都是直接从左向右匹配或者从右向左匹配，也有一些是找出最大的子串先进行匹配等，其中算法的一个关键点是在发现不匹配时，如何移动模式串使得比较时跳过尽可能多的字符以进行下一步的匹配。由于是字符串精确比较，所以两个单词必须长度相等，从这个角度进行初步筛选，可以排除大部分不匹配的单词。本算法对原文事先进行分词处理，使得比较时大大减少了匹配的字符数。

另外，由于很多单词部分字符相同，尤其是一些词根相同的，这就需要逐个字符比较，有的甚至比较到最后一两个字符才发现不是完全相同，这样比较浪费时间。而采用首尾字符筛选法，基本可以避免这种情况发生。

通过对多篇短文的统计，发现英文单词平均长度大约为 5~6,这就说明一般短文中长度在 5、6 个字符左右的英文单词较多。由以上实验也可以看出，单词长度越长的，用这种算法越节省时间，效率越高；相反，如果单词长度太短，则效率的提高不是非常明显。

4 结语

这种算法只适合于精确匹配，而模糊匹配正好相反，模糊匹配不要求精确的匹配结果,其目的是得到具有一定相似程度的匹配目标。例如查找一个单词的多种形式等。这种算法减少查询次数的程度和模式串的单词长度有关，一般模式串的单词越长越能体现出优越性。另外，这种算法只适用于分词比较清晰的文档进行检索，若文档单词间没有明显的间隔则不能直接使用这种算法。不过，可以通过对其他的算法(例如 BF 算法,ZZL 算法等)稍加改进，即先进性首尾字符比较，然后再比较中间字符，也可以节省很多匹配时间。

参考文献

1 杨薇薇,廖翔.一种改进的 BM 模式匹配算法.计算机应用, 2006,26(2):318－319.
2 陈开渠,赵洁,彭志威.快速中文字符串模糊匹配算法.中文信息学报, 2004,18(2):58－65.
3 王昕阳.浅析串模式匹配算法KMP及应用.电脑学习, 2007,(4):40－41.
4 王成,刘金刚.一种改进的字符串匹配算法.计算机工程, 2006,32(2):62－64.
5 潘景昌,孙玉辉,徐文明.一种简易的模糊匹配算法的实现.信号处理与模式识别, 2006,(2):131－132.