

一种改进的 Sunday 字符串匹配算法

王雁飞, 张文俊, 田 丰, 王海君
(上海大学 影视艺术与技术学院, 上海 200072)

摘 要: 在信息检索领域里, 字符串匹配是一个被广泛而深入研究的问题. 本文尝试分析几种字符串匹配算法的特点, 并以此为基础, 提出一种改进的 Sunday 算法. 该算法使用先转移后匹配的策略, 试图减少字符串匹配过程中无端的匹配. 实验表明, 本文改进的算法可以缩减匹配次数, 从而使匹配效率得到进一步提高.

关键词: 字符串匹配; Sunday 算法; 匹配方法; 模式识别

中图分类号: TP301.6

文献标识码: A

文章编号: 1000-7180(2013)04-0087-03

An Improved Method Based on Sunday String Matching Algorithm

WANG Yan-fei, ZHANG Wen-jun, TIAN Feng, WANG Hai-jun

(School of Film and TV Arts and Technology, Shanghai University, Shanghai 200072, China)

Abstract: In the field of information retrieval, the problem of string matching is researched widely and deeply. This paper analyzes the characteristics of several string matching algorithms. An improved Sunday algorithm based on the common string matching algorithms is presented. It attempts to reduce the number of unnecessary matching, using the idea of transfer before comparison. Experiments show that the algorithm effectively reduces the number of matching and enhances the matching efficiency.

Key words: string matching; Sunday algorithm; matching method; pattern recognition

1 引言

随着信息技术和互联网的飞速发展, 信息数据量越来越大, 在海量的数据中怎样能迅速检索到用户所要的数据, 是当今的较有意义的热点问题. 字符串匹配主要用于文本处理领域, 在一长文本串中, 找出其中一段较短的模式串. 目前, 字符串匹配的应用包括文本编辑、网络入侵检测、信息检索、模式识别、数据处理等. 在科学研究中, 字符串匹配除了在计算机领域应用较广, 在生物基因遗传学等领域也发挥着重要作用.

2 字符串匹配算法

首先定义文本串为 $T[1 \cdots n]$, 长度为 n ; 模式串为 $P[1 \cdots m]$, 长度为 m . 在字符串匹配算法中, BF 蛮力算法是最简单的一种算法, 先将模式串的左端与文本串的左端进行对齐, 若模式串与文本串不匹

配, 则将模式串相对文本串向右移动一位距离, 然后从模式串的最左端重新进行匹配. 由于该算法需要将文本串和模式串的索引值都往前回溯, 因此效率较低, 其算法复杂度为 $O(m \times n)$. Knuth 等人^[1]提出了一种 KMP 算法, KMP 算法在匹配效率上, 比 BF 算法有所提高, 但只是当文本串与模式串出现有较长的部分匹配片段时, 才体现出其优势. Boyer 等人^[2]提出一种 BM 算法, 在匹配过程中, 模式串仍按照从左往右的方向移动, 但字符的匹配却是从右端往左端逐个进行, 且制定了坏字符处理规则和好字符处理规则, 分别得到坏字符情况移动距离和好字符情况移动距离, 最后模式串的移动距离取坏字符情况移动距离和好字符情况移动距离中较大的数值, 从而使模式串获得尽量远的向右移动距离, 缩减匹配次数, 其计算复杂度为 $O((n-m) \times m + A)$, 字符集中的字符数用 A 表示, 最糟糕情况时计算复杂度为 $O(m \times n)$ ^[2]. Daniel M. Sunday^[3]提出一种

Sunday 字符串匹配算法,与 BM 算法相比,其算法思想简单,效率高,最糟糕情况时计算复杂度为 $O(m \times n)$ 。

近年来,研究人员大多都是对 KMP 和 BM 经典算法进行改进,以提高算法效率^[4]。本文在 Sunday 算法的基础上,结合文献[5]提出一种改进的先转移后匹配的算法,减少不必要的匹配次数,提高匹配效率。

3 Sunday 算法

首先定义文本串 $T[1 \cdots n]$,长度为 n ;模式串 $P[1 \cdots m]$,长度为 m ;文本串 T 中第 i 个字符用 $T(i)$ 表示,模式串 P 中第 j 个字符用 $P(j)$ 表示。

Sunday 算法的匹配过程是:当模式串 P 与文本串 T 不匹配时,即 $T(1 \cdots i)$ 不等于 $P(1 \cdots j)$,如图 1(a)所示。此时在模式串 P 中从最右往左依次寻找与字符 $T(i+1)$ 相等的字符,如果存在字符 $P(a)$ 等于字符 $T(i+1)$,则把模式串 P 向右移动 $j-a+1$ 的距离,使字符 $P(a)$ 与字符 $T(i+1)$ 对齐,然后继续判断 P 与 T 是否匹配,如图 1(b)所示。如果模式串 P 中找不到一个字符等于字符 $T(i+1)$,则把模式串 P 向右移动 $j+1$ 的距离,使得字符 $P(1)$ 与字符 $T(i+2)$ 对齐,然后继续匹配,如图 1(c)所示。

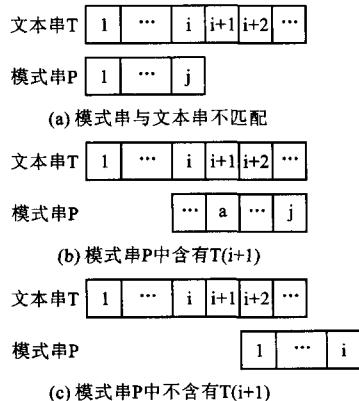


图 1 Sunday 算法的匹配方法示意图

4 改进的 Sunday 算法

由上述描述可知 Sunday 算法在运行时,先从首字符开始依次往后匹配比对,直到发现模式串与文本串不匹配时,才按 Sunday 规则进行跳转。文献[5]中采用让字符先转移再匹配的策略,使匹配次数得到缩减;该算法先判断模式串 P 中的末尾字符所在位置,对应到文本串 T 中相应的位置上的字符 $T(i)$ 是否在模式串中出现过,若没有出现则将模式串移动其自身的长度 m ,从字符 $T(i+1)$ 开始重新进

行判断;若 $T(i)$ 字符在模式串中出现,则将模式串进行跳转,使得模式串中最右端出现的与 $T(i)$ 字符相等的字符 $P(a)$ 与 $T(i)$ 字符对齐,然后从模式串最左端开始匹配,如果不匹配,则将模式串跳转距离增加一位,重新进行字符判断^[5]。

本文采用文献[5]中先转移后匹配的思想,并对文献算法中的跳转规则进行改进,算法关键步骤为:

①判断 $T(i)$ 字符是否在模式串中出现,出现则继续②,不出现转到③;

②将模式串进行跳转,使得字符 $P(a)$ 与字符 $T(i)$ 对齐,从模式串最左端开始判断模式串是否与文本串匹配,匹配则输出结果,并结束匹配过程,不匹配转到④;

③将模式串跳转 m 的距离,判断模式串是否已超出文本串,没有转到①,超出文本串则结束匹配;

④将模式串按照 Sunday 算法规则进行跳转,判断模式串是否已超出文本串,没有转到①,超出文本串则结束匹配;

本文改进算法的匹配过程流程如图 2 所示。

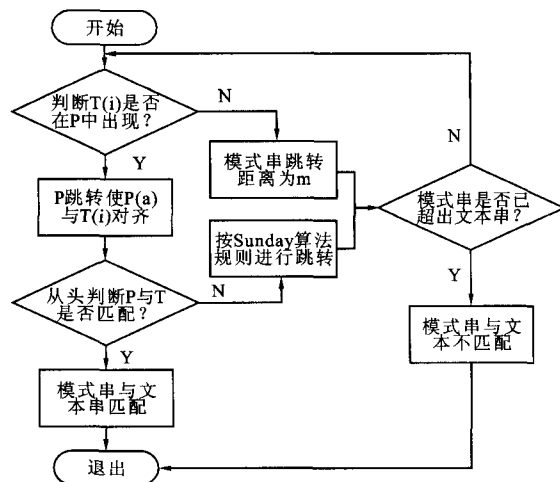


图 2 本文改进算法的匹配过程流程图

5 实验和结果分析

为了评测改进算法的有效性,本文在处理器为酷睿 2 双核,主频为 2.40GHz、内存为 2GB、操作系统为 32 位 Windows 7 的计算机上,利用 Matlab (R2011a)进行了多组实验比较。

实验 1 选取文献[5]中实验所采用的文本串 T 和模式串 P ,其文本串为“*For every incoming packet, we look up the class associated with its connection in the connection tracking unit, or, if it is a new connection, match the packet against all of the filters. If more than one filter matches, we as-*

sign it to the class with the highest priority. If no filter matches, the packet is discarded. To track connection cutoffs, the Time Machine keeps state for all active connections in a hash table. If a newly arrived packet belongs to a connection that has exceeded the cutoff limit configured for its class, it is discarded. We manage entries in the connection hash table using a user configurable inactivity timeout; the timeout is shorter for connections that have not seen more than one packet, which keeps the table from growing too large during scans or denial of service attacks.”模式串依次为 too, with, newly, exceeded, discarded. 分别利用本文算法、文献[5]中的算法和 Sunday 算法进行实验,比较各算法运行所需要的匹配次数. 本文同文献[5]一样,对算法的预处理过程不予考虑,而是将算法匹配过程中花费的匹配次数作为算法的评判指标^[5]. 实验结果如图3所示.

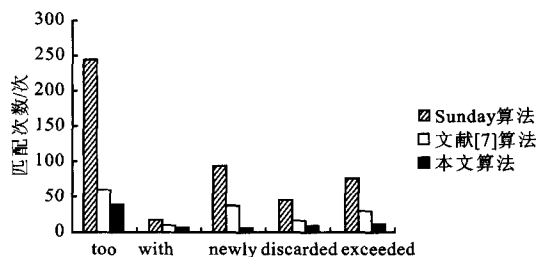


图3 各算法查找模式串所需匹配次数

实验2 选取文献[6]中实验所采用的文本串和模式串,进行实验,具体文本串和模式串见参考文献[6],实验结果如表1所示.

表1 本文算法和文献[6]算法的匹配次数比较

算法	Sunday 算法	文献[6]算法	本文算法
匹配次数	110	86	22

由图3、表1可直观的看出,在相同的文本串中搜索相同的模式串,相对于相应文献中的算法,本文改进算法使得搜索匹配次数得到进一步的减少. 本文算法采用先转移后匹配的策略,在模式串与文本串不匹配的情况下,采用 Sunday 算法的跳转规则,使模式串获得尽量远的移动距离,而不像文献[5]只移动一位,从而避免了更多无端的匹配,缩减了匹配次数.

6 结束语

针对字符串匹配问题,本文提出一种改进的 Sunday 算法,该算法在匹配过程中,采用先转移后匹配的策略,从而缩减无端的匹配次数. 小范围实验结果表明本文改进算法可使匹配次数得到有效缩减,使匹配效率得到进一步提高. 下一步研究方向将针对更全面的数据集对本文改进算法进行评估. 关于字符串匹配问题,算法预处理阶段的策略选择和时间效率也是影响算法整体效率的关键问题,还值得进一步研究.

参考文献:

- [1] Knuth D E, Morris J H, Pratt V R. Fast pattern matching in strings[J]. SIAM Journal on Computing, 1977, 6(2): 323-350.
- [2] R S Boyer, J S Moore. A fast string searching algorithm[J]. Communications of the ACM, 1977, 20(10): 762-772.
- [3] Daniel M. Sunday. A very fast substring search algorithm[J]. Communications of the ACM, 1990, 33(8): 132-142.
- [4] 王天聪,侯整风,何玲. 基于 BM 的模式匹配改进算法[J]. 合肥工业大学学报: 自然科学版, 2011, 34(3): 363-366.
- [5] 徐珊,袁小坊,王东. Sunday 字符串匹配算法的效率改进[J]. 计算机工程与应用, 2011, 47(29): 96-98.
- [6] 王成,刘金刚. 一种改进的字符串匹配算法[J]. 计算机工程, 2006, 32(2): 62-64.

作者简介:



王雁飞 男, (1987—), 硕士研究生. 研究方向为音乐检索、模式识别.

张文俊 男, (1959—), 教授, 博士生导师. 研究方向为数字媒体技术与应用、数字媒体、网络通信技术.

田丰 男, (1976—), 讲师, 博士. 研究方向为图像处理.

王海君 男, (1989—), 硕士研究生. 研究方向为真三维显示.