Computer Science and Application 计算机科学与应用, 2013, 3, 288-291
http://dx.doi.org/10.12677/csa.2013.36050 Published Online September 2013 (http://www.hanspub.org/journal/csa.html)

Study on the Searching Mongolian Based on String Matching Algorithms*

Hua Ju

Inner Mongolia Normal University Institute of Media, Huhhot
Email: 904859441@qq.com

Received: Jun. 16th, 2013; revised: Jul. 8th, 2013; accepted: Jul. 23rd, 2013

Abstract: Mongolian is an alphabetic writing. Its spelling rules are: the words are written vertically word by word, with a space between every two words, and the speech phonemes of a word are written together. Programmer always uses three string matching algorithms: the brute force string matching algorithms, the Boyer-Moore algorithm and the Horspool algorithm. When searching Mongolian key words, we should not only refer to the information searching technology of other languages, but also make some improvement according to the properties of Mongolian. This paper analyzes the string matching algorithms and the properties of Mongolian and improves the Horspool algorithm to search the Mongolian key words using six steps, and shows the selected status of the Mongolian key words in the corpus.

Keywords: Mongolian; Algorithm; String; Search

基于字符串匹配算法的搜索蒙古文的研究*

菊　花

内蒙古师范大学传媒学院，呼和浩特
Email: 904859441@qq.com

收稿日期：2013 年 6 月 16 日；修回日期：2013 年 7 月 8 日；录用日期：2013 年 7 月 23 日

摘　要：蒙古文是拼音文字，它的拼写规则是以词为单位竖写，词与词之间以空格分开，一个词的各个语音音素之间连写。常用的字符串匹配算法有蛮力字符串匹配算法、Boyer-Moore 算法和 Horspool 算法。实现蒙古文搜索时，不仅需要借鉴已有的其它语言的信息搜索技术，同时也需要依据蒙古文的特点进行改进。因此本研究通过对常用的字符串匹配算法及蒙古文的语法特点进行分析，改进 Horspool 算法，通过六个步骤，完成了从蒙古文语料中搜索相关关键词的任务，并在语料中以选中状态显示所搜索到的关键词。

关键词：蒙古文；算法；字符串；搜索

1. 引言

蒙古文是很早以前居住在我国北部和中亚地区的蒙古部落使用的语言，属于阿尔泰语系蒙古语族；蒙古文先后有六种，现行蒙古文有三种：传统蒙文(Mongolian)、托忒蒙文(Todo Mongolian)和新蒙文(Cyrillic Monglian)。传统蒙文是一种拼音文字，主要通行于我国内蒙古自治区，也是目前国际上通用的蒙古文字[1]。蒙古文字是具有自己特点的独特的拼音文字。它从左往右、自上而下书写，音节间无空格。一个蒙古文词都有词根，词根上加词缀可以构成一个新词。例如，“ ”的词根是“ ”，加词

*基金项目：教育部人文社会科学研究项目“现行蒙古文编码相互转换研究”(编号：10XJC740004)和内蒙古师范大学科技项目“基于语料的蒙古文单词分析软件的设计与实现(编号：KYZR1113)”。

缀“ ”后生成第二个词干“ ”，加词缀“ ”生成第三个词干“ ”。

蒙古文信息处理的工作始于 80 年代初，经过几十年的发展已有了很大进展。但蒙古文存在着一音多形，多音同形的现象。并且由于相关国际、国家标准的制定比较滞后，以及研究工作相对独立而各研究单位大都采用了各自的基于字型的蒙古文编码系统。而且用户对蒙古文输入法编码的掌握程度也不同，所以在蒙古文电子文本中普遍存在着字形正确、输入错误的现象。因此，对实现搜索蒙古文相关信息带来了许多困难，实现蒙古文搜索时不仅需要借鉴已有的其它语言的信息搜索技术，同时也需要依据蒙古文的特点进行改进。对于蒙古文信息处理，从蒙古文语料中快速搜索到相关关键词也很重要。

2. 随机文本中字符串匹配算法简介

在文本信息处理中，常涉及到单词的搜索的问题。常用的字符串匹配算法有蛮力字符串匹配算法、Boyer-Moore 算法和 Horspool 算法。

1) 蛮力字符串匹配算法

蛮力字符串匹配算法[2]中给定一个 n 个字符组成的串，称为文本，一个 m 个字符的串，称为模式，从文本中寻找匹配模式的子串。需要的是 i——文本中第一个匹配子串最左元素的下标——使得

$t_i = P_0, \cdots, t_{i+1} = pj,\ t_{i+m-1} = p_{m-1}$：

Text T $t_0 \cdots t_l \cdots t_{i+j} \cdots t_{i+m-1} \cdots t_{n-1}$

| | |

Pattern P P_0 P_l t_{m-1}

2) Boyer-Moore 算法

Boyer-Moore 算法[2]能把模式串 P 移动的距离更大，从而提高搜索的速度。此算法由四步完成所搜索的任务。

第一步：对于给定的模式和在模式及文本中用到的字母表。

第二步：按照给出的描述，利用模式来构造好后缀移动表。

第三步：将模式与文本的开始处对齐。

第四步：重复搜索的过程，直到发现了一个匹配子串或者模式到达了文本的最后一个字符以外。

3) Horspool 算法

Horspool 算法[2]中可以预先算出每次移动的距离并把它们存在表中。这个表是以文本中所有可能遇到的字符为索引的，对于一个自然语言的文本来说，这些

字符包括空格、标点符号和其他一些特殊字符。对于第一个字符 c，可以用以下公式算出移动距离：

$$t(c) = \begin{cases} \text{模式的长度m,如果c不包含在模式的前} \\ \text{m}-1\text{个字符中} \\ \text{模式前m}-1\text{个字符中最右边的c到模式} \\ \text{最后一个字符的距离} \end{cases}$$

此算法由以下几步来完成所搜索的任务：

第一步：对于给定的长度为 m 的模式和在模式及文本中用到的字母表，按照上面的描述构造移动表。

计算移动表中每个单元值的算法如下：初始时把所有的单元格都设置为模式的长度 m，然后从左到右扫描模式，将下列步骤重复 n − 1 遍：对于模式中的第 j 个字符($0 \le j \le m-2$)，将它在表中的单元格改写为 $m-1-j$，这是该字符到模式右端的距离。表中填充的移动距离是通过公式 1 计算。

第二步：将模式与文本的起始处对齐。

第三步：重复下面的过程，直到发现了一个匹配子串或者模式到达了文本的最后一个字符以外。从模式的最后一个字符开始，比较模式和文本中的相应字符，直到：要么所有 m 个字符都匹配(然后停止)，要么遇到一对不匹配的字符。在第二种情况下，如果 c 是当前文本中和模式的最后一个字符相对齐的字符，从移动表的第 c 列中取出单元格 t(c)的值，然后将模式沿着文本向右移动 t(c)个字符的距离。

3. 随机文本中字符串匹配算法分析

分析算法效率时有两种效率：时间效率和空间效率。时间效率指出正在讨论的算法运行得有多快；空间效率关心算法需要的额外空间。对于此算法重点考虑的是时间效率。对于随机文本中字符串匹配算法的时间效率分析如下[1]：

1) 对于蛮力字符串匹配算法它的最优时间效率 $C_{best}(n) = 1$,也就是模式 P 与文本 Text 进行比较时模式 P 不用移动就匹配；最差时间效率 $C_{worst}(n) = \theta(nm)$, 最坏情况下，在移动模式 P 之前算法可能会做足 m 次

比较，而 n − m +1 次尝试的每一次都可能会遇到这种情况；平均时间效率 $C_{avg}(n) = \theta(n + m) = \theta(n)$。

2) 对于 Boyer-Moore 算法，它的最差时间效率是线性的，该算法速度快。

3) 对于 Horspool 算法，它的最优时间效率 $C_{best}(n) = 1$，也就是模式 P 与文本 Text 进行比较时模式 P 不用移动就匹配；最差时间效率 $C_{worst}(n) = \theta(nm)$；平均时间效率 $C_{avg}(n) = \theta(n)$。

蛮力字符串匹配算法每次总是只移动一个位置，但 Horspool 算法可以预先算出每次移动的距离并把它们存在表中，所以平均来说，Horspool 算法显然要比蛮力算法快。Boyer-Moore 算法与 Horspool 算法比较速度更快，但如果模式最右边的字符和文本中的相应字符 c 所做的初次比较失败了，该算法和 Horspool 算法所做的操作完全一致，而且处理自然语言串的时候使用 Horspool 算法更简化，也更容易理解。

4. 基于字符串匹配算法的搜索蒙古文的方法

蒙古文是拼音文字。它的拼写规则是以词为单位竖写，词与词之间以空格分开，一个词的各个语音音素之间连写。总的书写规则是，蒙古文采取从上到下的顺序，由左到右的行序。蒙古文的词法结构较复杂，一个语音音素在词首、词中、词尾有三种不同的写法，同音不同形、同形不同音的现象很普遍。这些特点对于蒙古文信息处理带来了许多困难。通过上述对常用的字符串匹配算法分析，并依据蒙古文的语法特点，可以改进 Horspool 算法来完成从语料中搜索蒙古文的任务。比如输入一个蒙古文单词“ᠮᠣᠩᠭᠣᠯ”时，在多行文本框中显示出所搜索到的相关语料，并以选中状态显示，如图 1 所示。

基于字符串匹配算法，实现搜索蒙古文具体实现步骤如下：

1) 输入所搜索蒙古文时，可以选择一种蒙古文输入方法直接用键盘输入，也可以通过鼠标从所显示语料中选取的方法输入。本设计中选用了内蒙古明安途互联网技术开发有限公司开的蒙文 UNICODE IME 蒙古文输入方法。

2) 判断所输入的蒙古文单词的个数，并保存到变量 w-sum 之中。蒙古文词与词之间以空格或标点符号隔开，所以很容易实现所输入关键词的分词任务。

3) 构造移动表：所选用的输入方法在输入过程中，一个蒙古文单词的结束符为空格，因此以空格字符为索引构造出移动表，并根据变量 w-sum 来确定每一次的移动距离 t(c)。例如在语料中要搜索一个单词时，模式移动过程如图 2 所示。

实现移动表 ShiftTable(P[0…m-1])代码如下：

```
m = Len(Text1.Text) - 1'计算出模式长度
target = Chr(32)'Chr(32)代表为空格
For start_at = 1 To Len(Text2.Text) Step start_at
pos = InStr(start_at, Text2.Text, target) '搜索空格的位置
start_at = pos + 1'移动距离的改变
Next start_at
```

4) 将模式与文本的起始处对齐，即把初始变量 start_at 设置为 1。也可以通过改变 FindText()函数中

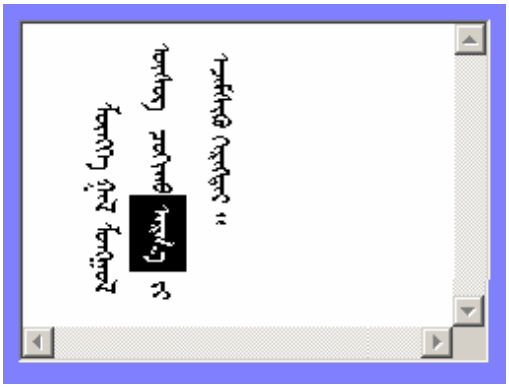

Figure 1. The result of searching a mongolian word
图 1. 搜索蒙古文单词结果

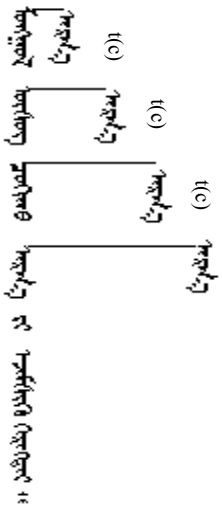

Figure 2. The process of moving pattern
图 2. 模式移动过程

的参数 start_at 的初始值来实现搜索出语料中出现的所有同一个词的功能。如果未找到所搜索的关键词，给出未找到的提示并给用户指出正确搜索的方法。

5) 每次计算出移动距离 t(c)后，要判断 m = t(c) − 1 是否成立，如果成立，再次判断模式是否匹配，如模式匹配就搜索成功，如不匹配继续重复上述操作，直到发现了一个匹配子串或者模式到达了文本的最后一个字符以外。

6) 选中的状态显示搜索到的蒙古文

搜索到的词在文本框的上下文中以选中的状态显示。用自定义的过程 AutoSelect 可完成以上所述功能。此函数有三个参数，第一个是控件类型，调用时传过来显示例句的控件。第二个参数接受所搜索词的开始位置，第三个参数接受所搜索词的长度。比如输入单词时，在多行文本框中显示出所搜索到的相关单词，并以选中状态显示，如图 1 所示。

由于蒙古文的特殊性，给编码和实现信息搜索等带来了一定的困难，因此能够高速有效地搜索蒙古文方面的工作还处在起步阶段。字符串匹配算法中，Horspool 算法比蛮力算法快，而且在处理自然语言中使用 Horspool 算法比 Boyer-Moore 算法更简化，更容易理解。因此依据蒙古文的语法特点，改进 Horspool 算法通过上述六个步骤来完成了从蒙古文语料中搜索相关关键词的任务[3-6]。

参考文献 (References)

[1] 清格尔泰. 蒙古语语法[M]. 内蒙古人民出版社, 1991.
[2] A. Levitin. Introduction to the design & analysis of algorithms. Pearson Eduction Inc, 2003.
[3] 斯•劳格劳, 敖其尔. Windows 环境下蒙古文复杂文本处理的研究[J]. 内蒙古大学学报(自然科学版), 2007, 5: 582-585.
[4] 巩政, 郝莉, 杨旭华. 非标准蒙古文字符编码转换为国际编码的一种方法[J]. 蒙古大学学报(自然科学版), 2008, 2: 216-219.
[5] 辛强. 基于共现距离与查询扩展的蒙古文信息检索系统[D]. 内蒙古大学硕士论文, 2011.
[6] 王成平. 彝文信息处理自动分词技术的研究现状与难点分析[J]. 电脑知识与技术, 2012, 2: 944-946.

