一种改进的基于后缀数组的无词典分词方法

刘京城,刘 锋
(安徽大学,安徽 合肥 230039)

摘 要:文中改进了基于后缀数组的无词典分词算法。原算法通过对输入字符集建立后缀数组并按字典序进行排列来筛选汉字结合模式形成候选词集,并通过置信度的比较来筛选候选词集以获得分词集。文中改进了其计算候选词出现频率的方法并且大大减少了筛选候选词集时两两判断候选词是否具有父子关系的次数。试验表明,改进的算法能够在没有词典的情况下更快速构建候选词集和筛选候选词集。适用于对词条频度敏感,对计算速度要求较高的中文信息处理。

关键词:自动分词;无词典分词;后缀数组
中图分类号:TP31 **文献标识码**:A **文章编号**:1673-629X(2011)11-0049-04

An Improved Automatic and Dictionary-Free Chinese Word Segmentation Method Based on Suffix Array

LIU Jing-cheng, LIU Feng
(Anhui University, Hefei 230039, China)

Abstract: It improved the original algorithm of automatic and dictionary-free Chinese segmentation based on suffix array. The original algorithm gets the candidate words by filtering the co-occurrence patterns of Chinese characters extracted from the input corpus with alphabetically sorted suffix array. And by filtering the candidate words through the confidence comparison the result set words are gotten. In this paper, improved the method that counted the frequency of the candidate words and reduced the number of judgments whether two candidate words have the father-and-son relationship when filtering the candidate words. Experiment results show that by the improved algorithm one can get and filter the candidate words more quickly without the help of the dictionary. This method is particularly suitable for lexical-frequency-sensitive as well as time-critical Chinese information processing application.

Key words: automatic word segmentation; dictionary-free word segmentation; suffix array

0 引 言

在当今这样一个信息爆炸的时代,如何迅速、有效地从大量数据中找到所需的信息已经成为信息服务领域中的重要问题。文本信息是人们用于信息记载和信息传播的最重要的媒体之一[1]。但是,汉语语言文字的处理与西文相比,有一个明显的不同就是汉语没有词与词之间的显式分隔标记[2]。所以中文文本信息处理的关键在于对文本中词汇的切分是否成功。

目前,国内研究学者们对中文文本的自动分词已进行了大量的研究,提出了许多自动分词方法,这些方法可以概括为三类:第一类是基于语法知识和规则的分词,但是由于汉语本身结构的复杂性,这种系统目前还处在试验阶段。第二类是基于词典的机械分词。该方法实现简单,实用性强,但缺点也很明显,就是词典的完备性无法得到保证。第三类就是基于统计的自动分词,该方法不需要词典的帮助,其核心原理是根据字符串在文本中的出现频率来决定其是否构成词[3-10]。文中采用的就是第三类方法,首先介绍张长利等人基于后缀数组的无词典分词算法[11];然后对他们的算法进行改进;最后通过实验来证明文中提出的改进算法能较大提高无词典分词的速度。

收稿日期:2011-03-22;**修回日期**:2011-07-04
基金项目:安徽省教育厅自然科学研究资助项目(KJ2009A60)
作者简介:刘京城(1986-),男,安徽黄山人,硕士研究生,研究方向为数据挖掘、软件工程;刘 锋,教授,硕士生导师,研究方向为软件工程与并行计算。

1 张长利等人基于后缀数组的无词典分词算法简介

1.1 数据结构

后缀数组是一种文本索引结构,这种结构记录了一字符串中各后缀的字典序索引。利用后缀数组和字符串编码可以对字符串集序列进行处理。近年来,在基因匹配、文本处理等领域中,后缀数组倍受关注。Manber 和 Myers[12] 给出了一个利用 $O(N)$ 的额外空间,在 $O(N\lg N)$ 时间内同时构造出后缀数组及最长公

共前缀信息数组 LCP 的算法[12,13]。

例 1 对于字符串“计算机与计算机程序及计算机程序设计”，计算其对应的后缀数组及最长公共前缀数组值如表 1 所示。

表 1 后缀数组及最长公共前缀数组对照表

SA[ ]	Value	Corresponding suffix	LCP	Value
SA[0]	3	与计算机程序及计算机程序设计	LCP[0]	0
SA[1]	9	及计算机程序设计	LCP[1]	0
SA[2]	8	序及计算机程序设计	LCP[2]	0
SA[3]	14	序设计	LCP[3]	1
SA[4]	2	机与计算机程序及计算机程序设计	LCP[4]	0
SA[5]	6	机程序及计算机程序设计	LCP[5]	1
SA[6]	12	机程序设计	LCP[6]	3
SA[7]	7	程序及计算机程序设计	LCP[7]	0
SA[8]	13	程序设计	LCP[8]	2
SA[9]	1	算机与计算机程序及计算机程序设计	LCP[9]	0
SA[10]	5	算机程序及计算机程序设计	LCP[10]	2
SA[11]	11	算机程序设计	LCP[11]	4
SA[12]	16	计	LCP[12]	0
SA[13]	0	计算机与计算机程序及计算机程序设计	LCP[13]	1
SA[14]	4	计算机程序及计算机程序设计	LCP[14]	3
SA[15]	10	计算机程序设计	LCP[15]	5
SA[16]	15	设计	LCP[16]	0

1.2 算 法

分词的目标是尽可能把文档中出现的词找出来，词的识别精度要尽可能高。

①取候选词集算法：见文献[11]GetWords( )；

②筛选候选词集算法：见文献[11]filterWords( )。

2 对以上算法的分析及改进

2.1 分析算法①和②

算法①的思路是将文档的所有后缀按字典序排列后，对于相邻两个后缀通过 SA[ ]和 LCP[ ]找出它们的最长公共前缀作为一个候选的词，并且将该最长公共前缀的所有从头开始的长度大于等于 2 的子串也作为候选词一并加入到候选词集 CandidateWords 中。比如以下两个相邻后缀：

计算机程序及计算机程序设计
计算机程序设计

其最长公共前缀“计算机程序”将作为候选词，并且其子串“计算”、“计算机”、“计算机程”都将作为候选词一并加入 CandidateWords。该算法在找出候选词时，同时也将其在文档中的出现频率也作为附加信息加入到 CandidateWords。

算法②的思路主要是根据置信度的比较结果来删除部分候选词。假定有两个串 W1 和 W2，它们在文档中的出现频率分别是 TF(W1)和 TF(W2)，并且 W2 是 W1 的子串，那么 W1 相对于 W2 的置信度就是 TF(W1)和 TF(W2)的比值，即 confidence(W1,W2)= TF(W1)/TF(W2)。设定置信度的上限和下限分别为 0.9 和 0.3，那么当 confidence(W1,W2)大于 0.9 时删除 W2，当 confidence(W1,W2)小于 0.3 时，删除 W1。最后再把剩下的候选词输出。整个算法的思路很简单，但其最大的缺点就是要对所有的候选词两两判断是否具有父子关系，而串包含关系的判断是个比较耗时的过程，对于大容量的文档，候选词的个数成千上万甚至几十万，两两判断的次数又比总个数要多出一个数量级，所以该过程将耗时巨大，无法满足用户需求。

2.2 算法改进

针对以上这种情况，文中提出一种新的思路，在取候选词时就对部分候选词进行置信度的比较，以避免到后来要对两词是否具有父子关系进行判断。

具体表述如下：在前面的取候选词算法①的第二层循环中，它是取该后缀的最长公共前缀的所有从头开始的全汉字子串作为候选词，所以此时很显然这些子串都被最长公共前缀所包含，因此可以直接对这些具有包含关系的候选词进行置信度的比较而无需再次判断它们是否具有父子关系，这样就可以为后面筛选候选词算法②节省大量的时间。还举例 1 的例子，从表 1 中看到倒数第二个后缀与其前一个后缀的最长公共前缀为“计算机程序”，它本身即为候选词，而且其子串“计算”、“计算机”、“计算机程”都为候选词，这些子串与其父都可直接比较置信度，而且“计算”、“计算机”与“计算机程”，“计算”与“计算机”也都可以直接进行置信度比较，无需先判断它们是否具有父子关系。虽然这里提前在取候选词时进行了部分候选词的筛选工作，增加了取候选词的时间，但它为后面单独地筛选候选词算法②省去了大量的判断是否具有父子关系的操作，为整个分词节省了大量时间。

实验验证结果请见表 2、表 3。具体算法如下：

算法③(改进后的取候选词集算法 getCandidateWords)：

输入：待分词的文档。

输出：文档的候选词集 CandidateWords(为散列表类型，包含每个候选词 key 及其在文档中的出现频率 value)。

getCandidateWords( )

{

1)初始化 CandidateWords 为空表，把文档读入到 corpusChar 字符串中；

2)利用算法[12]求得 corpusChar 字符串的对应后缀数组 SA[ ]与最长公共前缀信息数组 LCP[ ]。n 为 SA[ ]和 LCP[ ]的长度。假定 SA_i 为 SA[ ]中索引 i 对应的 corpusChar 的后缀(数组索引为 0 到 $n-1$)。

3)初始化 removeWords 为一个空的集合；//暂时

存放要被删除的候选词

4)for (i = 1; i < n; i++)

{

if (corpusChar 中第 SA[i]位置上的字符是汉字)//以汉字开头

{

(1)k= LCP[i];

(2)//将该后缀的最大公共前缀的从头开始的最大全汉字子串(以下将其作为最大公共前缀)的长度赋给 k

for (r = 1; r < k; r++)

if (corpusChar[SA[i] + r]不是汉字)

{

k = r;

break;

}

(3)新建长度都为 k+1 的字符串数组 keys[ ]、整型数组 values[ ]和布尔型数组 del[ ]并分别全部初始化为空、0 和 false; //由于要记录最大公共前缀及其子串的出现频率,所以设立临时数据结构,其中 keys[ ]存放候选词的 key,values[ ]存放候选词的 value,del[ ]指示该候选词是否要被删除

(4) // 从最大公共前缀的最末位开始往前搜索,将其所有从头开始的子串及其出现频率保存到 CandidateWords 以及上面建立的临时数据结构中

for (j = k; j≥ 2; j--)//j 表示词的汉字个数

{

1. key=corpusChar<SA[i],SA[i]+j - 1>;//取出 corpusChar 中第 SA[i]到 SA[i]+j - 1 的字符赋给 key,key 即为候选词

2. if(CandidateWords 已有 key)

{

keys[j] =key;//候选词

values[j] =key 对应的 value;//候选词的出现频率

}endif

else//CandidateWords 中没有 key,那就计算其出现频率

{

1. freq = 2;//freq 为 key 的出现频率的计数器,初始值为 2

2. p= i + 1;//p 为纵向指针

3. q= SA[p] + j - 1;//q 为横向指针

4. repeat

{

m=0;

repeat

{

if (corpusChar[SA [i] +1]! = corpusChar[SA [p] + 1])

break;//不相等则终止本层循环

m++;

}until(m≥j)

if (m ! = j) break;//说明该后缀不以 key 开头,故终止本层循环

freq++;

p= i + freq - 1;

q=SA [p] + j - 1;

}until(q≥n 或 p≥n)

5. value=freq;

6. 将(key,value)放到 CandidateWords 中;//key 为候选词,value 为 key 的出现频率

7. keys[j] = key;

8. values[j] = value;

}endelse

}endfor//j 从 k 到 2

(5)//根据临时数据结构,两两比较最大公共前缀及其所有从头开始的子串的置信度,然后对它们进行筛选

for (u = k; u > 2; u--)

for (v = u - 1; v > 1; v--)

{

if (del[v] 为 true) continue; //该子串已被删除,继续比较下一个子串

if (del[u] 为 true) break; //该串已被删除,无需再与其子串比较置信度

confidence = values[u] / values[v];

if (confidence > 0.9)del[v] = true;

if (confidence < 0.3)del[u] = true;

}

(6)//将要被删除的候选词暂时存放在 removeWords 集合中

for (t = 2; t < k + 1; t++)

if (del[t] 为 true) 则将 keys[t]加入到 removeWords 集合中;

// 注意,此时不能直接在 CandidateWords 中将 keys[t]候选词删除,因为在上面第(4)点的循环体第 2. 步中需要根据 CandidateWords 是否已经有 keys[t]候选词来决定其出现频率如何获得(不同的后缀有可能找到相同的候选词)

}endif//SA[i]位置是汉字

}endfor//i 从 1 到 n-1

5)对于 removeWords 中存放的每一个候选词,如果在 CandidateWords 也有的话就将其删除;//取 CandidateWords 与 removeWords 的差集

6)输出 CandidateWords.

}

3 实验结果及分析

文中搜集的几份实验材料都是大家比较熟悉的小说的电子版(txt 格式的文件,并且按小说字数从小到大排列),用这些材料对算法的改进成果进行了验证。实验结果如表2和表3所示。

表2 算法③改进效果参照表(一)

书名	使用算法①时算法②父子关系判断次数	使用算法③时算法②父子关系判断次数	算法③减少父子关系判断次数百分比
活着	91305439	48033660	47.39%
芙蓉镇	198454372	128699810	35.15%
围城	364561220	212515985	41.71%
山楂树之恋	460441348	191702867	58.37%
大宅门	1335283398	519844807	61.07%
平凡的世界	8224050311	2498087715	69.62%
天龙八部	12129452884	4066750905	66.47%

表3 算法③改进效果参照表(二)

书名	算法①耗时(秒)	算法③耗时(秒)	算法①+②耗时(秒)	算法③+②耗时(秒)	速度提升倍数
活着	0.015625	0.0625	19.609375	7.28125	1.69
芙蓉镇	0.09375	0.109375	49.984375	21.625	1.31
围城	0.125	0.140625	93.671875	32.09375	1.92
山楂树之恋	0.109375	0.1875	92.59375	32.734375	1.83
大宅门	0.203125	0.421875	301.34375	102.03125	1.95
平凡的世界	0.9375	1.140625	2794.078125	549.09375	4.09
天龙八部	0.796875	1.390625	2938.109375	897.484375	2.27

从表2实验结果可以看出,使用改进的算法③后大幅度缩减了父子关系的判断次数,而且随着文档容量(字数)的增加缩减的百分比也增加。对比表3中各材料的算法①与算法③耗时可看出,算法③与①相比多出来的比较置信度的操作所消耗的时间非常微小,可是由于使用算法③,算法②消耗的时间大幅减少,整个分词过程的提升倍数也在1倍多到4倍多不等,而且从直观上看也有随文档容量增加提速倍数上升的趋势,说明算法③的改进效果确实较好。

4 结束语

文中在深入分析张长利等人基于后缀数组的无词典分词算法[11]的基础上对其分词速度进行了改进,主要在于减少其找候选词出现频率的时间及对候选词父子关系的判断次数以达到提高分词速度的目的。对于其分准率和分全率文中未能做进一步的改进,在筛选候选词时置信度上下限的设置方面以及多加入一些助动词以帮助筛选方面都还有优化的空间,有待进一步研究。

参考文献:

[1] 邹 涛,王继承,杨文清,等. 文本信息检索技术[J]. 计算机科学,1999,26(9):72-75.

[2] 邹海山,吴 勇,吴月珠,等. 中文搜索引擎中的中文信息处理技术[J]. 计算机应用研究,2000(12):21-24.

[3] 陈桂林,王永成,韩客松,等. 一种改进的快速分词算法[J]. 计算机研究与发展, 2000,37(4):418-424.

[4] 付国宏,王晓龙. 汉语词语边界自动划分的模型和算法[J]. 计算机研究与发展, 1999,36(9):1142-1147.

[5] 吴胜远. 一种汉语分词方法[J]. 计算机研究与发展,1996,33(4):306-310.

[6] 黄德根,杨元生,王 省,等. 基于统计方法的中文姓名识别[J]. 中文信息学报,2001,15(2):31-44.

[7] 刘开瑛,薛翠芳,郑家恒,等. 中文文本中抽取特征信息的区域与技术[J]. 中文信息学报, 1997,12(2):1-7.

[8] 黄德根,朱和合,王昆仑,等. 基于最长次长匹配的汉语自动分词[J]. 大连理工大学学报, 1999,39(6): 831-835.

[9] 傅赛香,袁鼎荣,黄柏雄,等. 基于统计的无词典分词方法[J]. 广西科学院学报, 2002,18(4): 252-255.

[10] 胥桂仙,苏筱蔚,陈淑艳. 中文文本挖掘中的无词典分词的算法及其应用[J]. 吉林工学院学报, 2002,23(1): 16-18.

[11] 张长利,赫枫龄,左万利,等. 一种基于后缀数组的无词典分词方法[J]. 吉林大学学报, 2004,42(4):548-553.

[12] Udi M, Myers G. Suffix arrays: a new method for on-line string searches [J]. SIAM Journal on Computing, 1993, 22(5):935-948.

[13] Yamamoto M, Church K. Using suffix arrays to compute term frequency and document frequency for all substrings in a corpus[J]. Association for Computational Linguistics, 2000, 27(1):1-30.

(上接第48页)

Administration Review,1994(2):54-59.

[11] 何代欣,罗 为,王 谦. 基于电子政务的绩效评估方法的构建与实施[J]. 广西财政高等专科学校学报,2005,18(5):57-59.

[12] 杨卫平,段丹青,黄烟波. 警务绩效的模糊综合评价方法研究[J]. 计算机技术与发展,2007,17(6):228-231.

[13] 曾勇进,庄建东,李 思. 电子政务评估方法 AHP 的研究及实现[J]. 心智与计算,2010,4(1):55-62.

[14] 申小蓉,汪 洁. AHP 法在城市政府管理评估指标体系中的应用[J]. 电子科技大学学报,2007,36(1):154-157.

[15] 耿兴荣,林炳耀. 中国信息化评价指标体系初探[J]. 经济地理,2002,22(6):724-729.