基于后缀数组的分词技术①

任雪利 代余彪 (曲靖师范学院 计算机科学与工程学院 云南 曲靖 655011)

摘 要: 中文分词技术是机器翻译、分类、搜索引擎以及信息检索的基础，但是，互联网上不断出现的新词严重影响了分词的性能，为了提高新词的识别率，建立待分词内容的后缀数组，然后计算其公共前缀共同出现的次数，采用阈值对其进行过滤筛选出候选词语，实验结果表明，该方法在新词识别方面有一定的优势。

关键词: 后缀数组;分词;公共前缀长度

Word Segment Based on Suffix Array

REN Xue-Li, DAI Yu-Biao

(Department of Computer Science and Engineering, Qujing Normal University, Qujing 655011, China)

Abstract: Chinese word segmentation technology is the basis of machine translation, classification, search engines, as well as information retrieval. But the Internet emerging new words have seriously affected the performance of word segmentation. To improve the recognition rate of new words, suffix array is used in this paper, and the number of length of common prefix is calculated. The candidates on their words are filtered out by the threshold. Experimental results show that the new word recognition method has advantages.

Keywords: suffix array; word segment; LCP

由于中文是以词为单位，但是在中文文本中词与词之间没有明确的边界，因此，需要对中文文本进行处理，划分出有独立意义词，该过程称为分词。并且中文分词技术是机器翻译、分类、搜索引擎以及信息检索的基础。互联网的快速发展极大的影响着人类的生活，不断出现的新词语严重影响了机器翻译、分类及信息检索的准确性，因此，如何有效的识别出新词也成为当今分词技术必须考虑的问题，为了解决分词过程中的新词识别率，本文采用后缀数组的分词技术。

1 分词技术及后缀数组简介

1.1 分词方法简介

现有的分词算法可分为三大类: 基于字符串匹配的分词、基于理解的分词和基于统计的分词。基于字符串匹配的方法包括正向匹配法、最短路径法等。字符串匹配的分词方法利用已经建立起来的电子词典对输入文本进行简单匹配，若在词典中找到某个字符串则匹配成功。该方法的优点是易于实现，但精度较低。基于理解的分词方法其基本思想就是在分词的同时进行句法、语义分析， 利用句法信息和语义信息来处理歧义现象，这种分词方法需要使用大量的语言知识和基础信息资源规则维护困难，但是切分准确率高。基于统计的分词方法通过对语料中相邻共同出现的各个字的组合频度进行统计，计算它们的互现信息，互现信息体现了汉字之间结合关系的紧密程度，这种方法只需对语料中的词组频度进行统计，能够较高效地识别未登录词 但是此方法需要大量切分词典的原始文档，且计算量大。如果提取结果中存在意义不完整

① 基金项目:曲靖师范学院基金(2008QN007);云南省教育厅研究课题(09C0188)
收稿时间:2009-12-04;收到修改稿时间:2010-01-18

的字符串，则导致准确率不高。

1.2 后缀数组

后缀数组是一种紧凑的数据结构，可处理任意长度的字符串并且实现功能强大的搜索。受到 PAT 树的支持，但是后缀数组使用的空间更少。下面给出几个重要的定义[1-4]：

定义 1．S'表示在字符集Σ上的一个字符串，$Σ是唯一的终结符且小于字符集Σ中的任一字符，LS=S'$是在字符串S'的末尾加上终结符得到的一个新字符串，称为 S 的左后缀数组 LS;如果|S|表示字符串的长度，LS[i]表示 S 的第 i 个字符，那么 $Lsuff_i$=S[i]S[i+1]…S[|S|]是字符串 S 的第 i 个后缀数组。例如：S'=“asdfgh”，在其后增加一个结束符$，得到 LS=“asdfgh$”，那么，LS_2=“dfgh$”是 S 的第 2 个左后缀数组。同理可以构造右后缀数组 RS= $S'是在字符串 S'的首部加上终结符得到的一个新字符串，RS[i]表示 S 的第 n-i 个字符，那么 $Rsuff_i$=S[n-i]S[n-i-1]…S[0]是字符串 S 的第 i 个后缀数组。例如：S'=“asdfgh”，在其最前端增加一个结束符$，得到 LS=“$asdfgh”，那么，LS 2=“fdsa$”是 S 的第 2 个右后缀数组。

定义 2．公共前缀是指后缀字符串按首字符排序后，两个相邻的字符串从左向右数相同的字符个数。左公共前缀记为 LLCP，右公共前缀记为 RLCP。

下面以字符串”upcfpsopcf”为例说明后缀数组的概念。

1)构造后缀数组

	0	1	2	3	4	5	6	7	8	9	10	11
	$	u	p	c	f	p	s	o	p	c	f	$

2) 按首字母对后缀数组进行排序，结果如下：

	0	1	2	3	4	5	6	7	8	9	10	11
左	0	11	3	9	10	4	7	8	2	5	6	1
右	0	11	9	3	4	10	7	5	2	8	6	1

3) 计算公共前缀长度

	0	1	2	3	4	5	6	7	8	9	10	11
LLCP	0	0	2	0	1	0	0	3	1	0	0	/
RLCP	0	0	2	0	3	0	0	1	1	1	0	/

2 基于后缀数组的分词方法

词是汉语的基本构成单位，分词方法根据在分词的过程中是否使用字典分为字典分词和无字典分词，对于字典分词，根据待分词的文本中取出词然后在字典中查找，若该词在字典中出现，则它就是词；无词典的分词，根据在一篇文档中重复出现的内容作为词。张长利等提出了一种无字典的分词方法[5]，该方法通过对全文建立后缀数组，然后将重复的内容进行聚集，如果该内容出现的次数大于设定的阈值，则认为它是一个词，虽然该方法可以实现无字典的分词，但是可能将重复出现但不是词的内容作为词，例如在《笑傲江湖》这篇小说中，“与东方不败”出现了很多次，那么就被作为一个词，事实上“东方不败”是一个词，而“与东方不败”并不是一个词；并且该方法将整篇文本作为一个字符串来建立后缀数组，虽然建立后缀数组的方法简单，但是对内存的需求是非常大的；例如对于一个包含 1000 个字符的文本，建立后缀数组后需要的空间大小为 1000+999+998+…+1=500500 字节，也就是原文本的 50 多倍，且若文本的大小为 n 字节，建立后缀数组后所需的空间为 n(n+1)/2 字节，鉴于该方法存在不准确和空间需求大的问题，本文通过从两个方向对文本进行分词，然后取其共同的部分作为词，这样可以提高分词的准确性并且降低对内存空间的需要。下面给出基于后缀数组的分词方法的具体步骤：

1)建立文本的左右后缀数组 LS 和 RS。将整篇文档按照标点符号进行分割生成不同的句子，然后对每一句话建立其左右后缀数组 LS 和 RS；

2)按照首字对文本进行排序。根据建立的左右后缀数组 LS 和 RS，分别按照首字符的字典顺序对其进行排序；

3)计算公共前缀的长度。对排序后的左右后缀数组，分别计算其公共前缀的长度 LLCP 和 RLCP；

4)筛选出候选词语。将公共前缀中数值大于等于 2 且连续出现次数大于 3 的公共前缀筛选出来，并根据公共前缀的值 m 从头开始取 m 个字作为候选的词。

5)筛选词。对照左右后缀数组候选词，找出含有共同的多于 1 个字的部分作为词，然后删除含有该词的候选词。

2.1 算法分析

1)本方法通过从两个方向建立后缀数组，然后从

(下转第 211 页)

(上接第 230 页)

两个方法分别找出高频词作为候选，然后将左右候选词进行对照，取出公共的部分作为一个词，这样可以去掉词语前后的停用词，识别出真正的词；

2)若文本的大小为 n 字节，假定平均 10 个字节为一句话，那么，文本包含 n/10 个句子，每一句子建立后缀数组需要 1+2+...+10=55 个字节建立后缀数组，那么共需要 55n/10 来建立一个方向的后缀数组，该方法共需要 55n*2/10=11n 字节的内存空间，也就是说建立后缀数组需要的空间是原文本规模的 11 倍，与原文本的规模没有关系。

3 实验结果

本文采用 C++语言对不同领域、规模递增的 5 个(A,B,C,D,E)中文文本进行分词，使用准确率和召回率作为分词结果的评价标准，实验结果如图 1 所示：

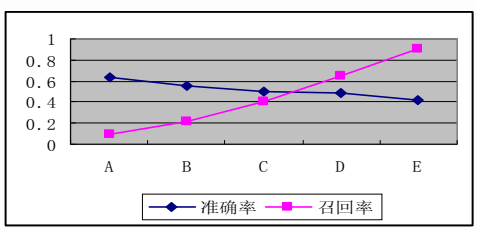

图 1 分词结果图

由图 1 可知：随着文本规模的增大，识别出的新词相应的增加，因此，召回率增大；准确率随着文本规模的增大有略微的降低，是由于有些字经常一起出现，例如：“这个”、“这一“、“说的“等，但是并不是词而引起的。

参考文献

1 Mcilroy MD. A Killer Adversary for Quicksort Software-Practice and Experience,1999,29:5－12.

2 Larsson NJ, Sadakan K. Faster suffix sorting. LU-CS-TR,1999.99－214.

3 Manber U, Myers G. Suffix Arrays: A New Method for On-Line String Searches, SIAM Journal on Computing, 1993,22:10－11.

4 Lee feng Chien,PAT-Tree-Based Keyword Extraction for Chinese Information Retriveal.[2009-10-15].http://citeseerx.ist.psu.edu/viewdoc/download.

5 张长利.一种基于后缀数组的无字典分词方法.吉林大学学报(理学版),2004(42):548－553.