基于汉语拼音首字母索引的混合分词算法①

杨进才[1], 陈忠忠[1], 谢 芳[2], 胡金柱[1]

[1](华中师范大学 计算机学院, 武汉 430079)
[2](湖北工业大学 计算机学院, 武汉 430068)

摘 要: 中文自动分词是web文本挖掘以及其它中文信息处理应用领域的基础. 蓬勃发展的中文信息处理应用对分词技术提出了更高的要求. 提出了一种新的分词算法 FPLS, 该算法用拼音首字母作为词语表一级索引, 词语的字数为二级索引构造分词词典, 采用双向匹配方法, 并引入规则解决歧义切分问题. 与现有的快速分词算法比较, 该算法分词效率高且正确率高.

关键词: 中文分词; 拼音索引; 双向匹配; 歧义切分

Hybrid Segmentation Algorithm for Chinese Text Using First Pinyin Letter Index

YANG Jin-Cai[1], CHEN Zhong-Zhong[1], XIE Fang[2], HU Jin-Zhu[1]

[1](School of Computer Science of Central China Normal University, Wuhan 430079, China)
[2](School of Computer Science of Hubei University of Technology, Wuhan 430068, China)

Abstract: Chinese automatic segmentation is the basis of web text mining and other Chinese information processing applications. Booming Chinese information processing applications put forward a higher requirement for Chinese automatic segmentation. This paper presents a new segmentation algorithm FPLS, which uses a dictionary with a first letter of the Pinyin as a first level index and words count as the secondary index structure. A bidirectional matching method and rules are employed to resolve ambiguity segmentation problem in the algorithm. Comparing with the existing algorithm, algorithm FPLS gets higher accuracy and efficiency.

Key words: Chinese automatic segmentation; Pinyin index; bidirectional match; ambiguity resolve

自然语言人机接口、情报检索、web 查询系统、文本数据挖掘以及应用最广泛的搜索引擎的研究均依赖于中文信息处理的研究. 在中文信息处理研究中自动分词算法是基础课题, 应用环境不同对自动分词要求也有所不同. 有一些对于速度要求非常高, 如处理海量数据的搜索引擎, 有一些对于精度的要求比较严格, 如自然语言的理解、自动翻译等.

自动分词算法研究的主要容是设计高效的词表数据结构以及算法, 以满足不同的分词要求. 在中文信息处理领域的高速发展的 20 年来, 许多专家、学者提出了不同的自动分词算法如: MM 方法[1,2]、多次 Hash 快速分词算法[3]、全二分查找算法[4]、双哈希二叉树分词算法[5]、规则的分词算法[6]、词频的分词算法[7]等.

这些分词算法归为三大类: 机械分词方法、基于统计的分词方法和基于规则的分词方法. MM 方法、多次 Hash 快速分词算法、全二分查找算法和双哈希二叉树分词算法属于机械分词方法. 规则的分词算法属于基于理解的分词方法. 词频的分词算法属于基于统计的方法. 机械分词方法无法解决分词阶段的歧义切分问题和未登录词识别问题, 使用过程中需要借助其他的信息提高精确度. 基于规则的分词方法对信息的提取较难, 因此对其研究还处在试验阶段. 基于统计的分词方法需要使用词频度. 它不仅考虑了句子中词语出现的频率信息, 同时也考虑到词语与上下文的关系, 具备较好的学习能力, 对歧义词和未登录词的识别有

① 基金项目:教育部社科基金(13YJAZH117);国家社科基金(14BYY093)
收稿时间:2015-07-28;收到修改稿时间:2015-09-21

良好的效果[8]. 但它也有一定的局限性, 会抽出一些出现频度高、但并不是词的常用字组.

随着大数据时代的到来, 海量的文本信息需要中文分词既准确, 同时快速. 本文将探讨一种新的分词算法, 在优先保证高速的同时提高分词的准确率.

1 基本自动分词算法

基本的自动分词算法有两种: 正向匹配算法与逆向匹配算法.

1.1 正向最大匹配分词算法

正向最大匹配分词算法是一种应用最为广泛的机械分词算法, 这种算法又叫最长匹配法、回巡检索法, 本文称正向匹配算法. 算法描述如下:

假设自动分词词典中的最长词条含有 n 个汉字

(a)输入要处理的字符串, 取前 n 个字为匹配字段.

(b)对匹配字段查找分词词典, 如果匹配成功, 匹配字段作为一个词就被切分出来;如果查不到, 去掉匹配字段的最后一个字, 剩余的 n-1 个字再作为匹配字段进行匹配, 直到字段匹配成功.

(c)将句子中剩下的部分作为匹配字段, 重复进行步骤(a)(b)(c)直至匹配完成为止.

1.2 逆向最大匹配分词算法

逆向分词算法与正向分词算法大体相似, 只是匹配从后往前进行, 而且使用的词典也不相同, 它使用逆序词典, 其中每个词语以逆序的方式存放. 匹配不成功去掉前面一个字继续匹配直至匹配成功.

1.3 正向分词算法与逆向分词算法的分析对比

单纯的使用逆向最大匹配分词算法的错误率为 1/245, 单纯的使用正向最大匹配分词算法的错误率为 1/169[9]. 逆向最大匹配分词算法准确率要比正向最大匹配分词算法的高.

例 1: "老师讲不到的学生会学."

正向最大匹配分词算法会切分成"老师\讲\不到\的\学生会\学",逆向最大匹配分词算法会切成"老师\讲\不到\的\学生\会学". 分析正向匹配和逆向匹配的错误, 会发现错误的部分大多数是正向匹配与逆向匹配不一致即出现歧义.

例 2: "老师讲的题学生会"

正向最大匹配分词算法会切成"老师\讲\的\题\学生会", 逆向最大匹配分词算法会切成"老师\讲\的\题\学生会"此时, 虽然正向最大匹配分词算法与逆向最大分词算法的结果一致, 但是同样出现了歧义. 正确的结果应为"老师\讲\的\题\学生\会"

通过分析例 1 和例 2, 我们发现正向最大匹配和逆向最大匹配结果不一致和一致都有可能出现歧义. 解决好这些歧义可以提高分词的正确率.

2 拼音首字母分词词典

2.1 词库构造

汉语中的词是最小的、独立的, 有重要意义的语言成分, 是组成语言的最小单位. 汉语中词是一个开放的集合, 数量是无穷的, 但可收集的词却是有限的. 常用的词有 43570 个, 这些词的长短有所不一, 从短到一个字到长到七个字的均有, 其中二个字词最多. 具体分布如表 1 所示.

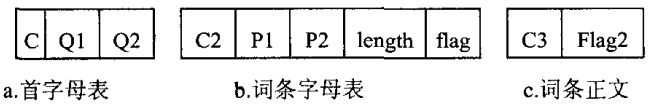
表 1　词库词条字数分布表

词条字数	1	2	3	4	5	6	7
词数	2606	33527	3693	3622	83	36	3

组成词的汉字虽很多, 但拼音却只有 496 个(在不考虑音调的情况下), 而对应的首字母更少仅有 26 个(a-z). 如果按汉语拼音的首字母划分词条, 就会将词条划分成 26 个部分.

2.2 分词词典

分词词典是组成汉语自动分词系统的重要成分, 汉语自动分词系统需要从分词词典中提取信息. 这里设计一种拼音首字母分词词典, 词典分为三部分: 首字母表、词条字母表、词典正文. 词典的结构如图 1 所示.



a.首字母表　　b.词条字母表　　c.词条正文

图 1　词典结构图

(1) 首字母表

每一个汉字在首字母表中都能找到唯一的缩写字母与其相照应, 首字母表中每一项的结构如图 1a 所示.

其中, C 为首字母; Q1 为指向在词条字母表中第一次出现以 C 开头的字符串的指针; Q2 为指向在词条字母表中最后一次出现以 C 开头的字符串的指针.

例如, 若 C 为'a', 则 Q1 指向词条字母表中的"a", Q2 指向词条字母表中的"alpkydh"(其中"alpkydh"为在

词条字母表中最后一次出现且以'a'为开头的字符串).

(2) 词条字母表

词条字母表对应字典正文中唯一的一项. 其中, C2 为拼音首字母所组成的字符串; P1 为指向词典正文中第一次出现拼音首字母缩写为 C2 的词语的指针; P2 为指向词典正文中最后一次出现拼音首字母缩写为 C2 的词语的指针; length 为 C2 的长度; flag 为是否有与待查询的缩写字母相匹配的C2的标志, 初始值设为 0.

在图 2 中, 若 C2 为"a", 则 P1 指向"啊"的指针, P2 为指向"奥"的指针, length=1.

(3) 词典正文

词典正文是由词构成.其中, C3 为词语; flag2 是否匹配成功的标志词, 初始值设为 0.

将上述三的结构构成分词词典整体结构图如图 2 所示.

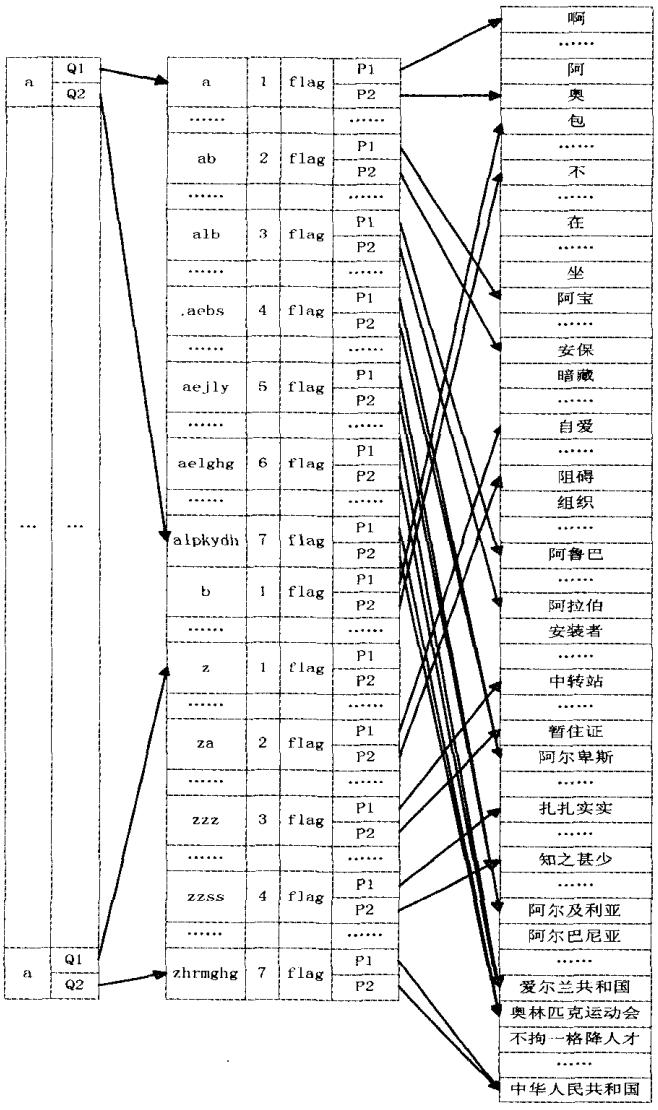


图 2 分词词典整体结构图

3 分词歧义字段处理

不同的分词方法对同一段文本进行分词, 结果可能不相同, 其中不同的部分称为歧义字段.

例如: "我们出现奥运会场. "

正向最大匹配算法会切成"我们\出现\奥运会\场", 而逆向分词算法会切成"我们\出现\奥运\会场". 其中"奥运会"和"会场"出现歧义, "会"为交集字串. 处理分词中出现的歧义字段, 是分词中的一个难点. 歧义字段的类型有交叉型歧义字段与组合型歧义字段两种. 交叉型歧义字段指 A、B、C 三个子串, AB、BC 分别构成词则有两种切分方式:AB/C 和 A/BC, 而组合型歧义字段指对于汉字串 AB 既可切分为 AB 又可切分为 A/B[10]. 歧义字段中交叉型最多, 交叉型歧义字段占全部歧义字段的 94%[11]. 为了解决分词中出现的歧义问题, 本文在 PFLS 算法的基础上采用规则进行消歧. 设计规则如下:

(1) 交集字串与其后继的字串构成形容词, 将歧义字段的首字切掉. 如: "太美好", 交集字串为"美", "美"与后继构成形容词将其前驱切掉, 结果为"太\美好".

(2) 交集字串的前驱为数词, 将歧义字段的首字切掉. 如: "三个人"交集字串为"个", "三"为数词将其切掉, 结果为"三\个人".

(3) 交集字串与后继构成动词且与前驱也构成动词, 将尾字单切. 如: "骚扰乱民" 交集字串为"扰", "扰乱"为动词将"乱"切掉, 结果为"骚扰\乱民".

(4) 歧义字段类似 ABC\D, 交集字串与后继构成动词且前驱为名词, 将前驱切掉. 如: "老师不讲, 学生会学"交集字串为"会", "学生"为名词, "会学"为动词切掉前驱, 结果为"老师\不讲\, \学生\会学".

(5) 交集字串与后继构成名词且前驱为动词, 将前驱切掉. 如: "劳动力气"交集字串为"力", "力气"为名词, "劳动"这里为动词切掉前驱, 结果为"劳动力气".

(6) 交集字串的后继为助词, 将尾字切掉. 如: "她是一个娇小的女孩"交集字串为"小", 有"娇小"和"小的"两种切分方式, "的"为助词, 结果为"她\是\一\个\娇小\的\女孩".

(7) 交集字串的后继为后接成分, 将尾字切掉. 如: "大人们", "人"为交集字串, "们"为后接成分将其切掉, 结果为"大人\们".

4 算法描述

将这种按拼音首字母分词的分词算法称为 FPLS (First Letter of the Pinyin Segmentation), 算法描述如下:

(1) 基于拼音首字母的正向匹配算法

算法名: *FR ForeMatch*(Text)

输入: Text 为一段文本

输出: FR, 为一重复有序集合, 集合中的元素为字与词.

(a)将 *Text* 保存于数组 *s*1 中,将 *Text* 汉字转化为拼音首字母存储于数组 *s*2 中;

(b)字符个数 *n*=7;

(c) 取 *s*2 中前 *n* 个字符 *s*, 先根据首字符在首字母表中查找. 在首字母表中找到与之对应的 *C* 之后, 从 *Q*1 指向的字符串开始, 在词条字母表中逐个匹配, 直至 *Q*2 所指向的字符串为止.

(d)在词条字母表中未找到与 s 相匹配的字符串, *flag*==0, *n*=*n*-1(去掉 *s* 的最后一个字符)重新进行(c)步骤; 找到与 *s* 相匹配的字符串, *flag*==1, 从 *P*1 指向的词语开始, 在词典正文中逐个匹配 *s* 对应的词, 直至 *P*2 所指向的词语为止.

(e)在词典正文中未找到匹配的词语, *flag*2==0, 去掉 *s* 的最后一个字符重新进行(c)(d)(e)步骤. 若在词典正文中找到匹配词语, *flag*2==1, 将 *s* 对应的 *s*1 中的词作为一个词切分, 并将结果加入 *FR* 中.

(f)重复(b)(c)(d)(e)步骤, 直至 *s*1 全部切分完毕.

(2) 基于拼音首字母的逆向匹配算法

算法名: *BR BackMatch*(*Text*)

输入: Text 为一段文本

输出: BR, 为一重复有序集合, 集合中的元素为字与词

(a)将 *Text* 保存于数组 *s*1 中,将 *Text* 汉字转化为拼音首字母存储于数组 *s*2 中;

(b)字符个数 *n*=7;

(c)取 *s*2 中后 *n* 个字符 *s*, 先根据首字符在首字母表中查找. 在首字母表中找到与之对应的 *C* 之后, 从 *Q*1 指向的字符串开始, 在词条字母表中逐个匹配, 直至 *Q*2 所指向的字符串为止.

(d)在词条字母表中未找到与 s 相匹配的字符串, *flag*==0, *n*=*n*-1, 去掉 *s* 的最前一个字符)重新进行(c)步骤; 找到与 *s* 相匹配的字符串, *flag*==1, 从 *P*1 指向的词语开始, 在词典正文中逐个匹配 *s* 对应的词, 直至 *P*2 所指向的词语为止.

(e)在词典正文中未找到匹配的词语, *flag*2==0, 去掉 *s* 的最前一个字符重新进行(c)(d)(e)步骤. 若在词典正文中找到匹配词语, *flag*2==1, 将 *s* 对应的 *s*1 中的词作为一个词切分, 并将结果加入 *BR* 中.

(f)重复(b)(c)(d)(e)步骤, 直至 *s*1 全部切分完毕.

(3) 基于拼音首字母的双向匹配算法

执行(1)(2)得到 FR 和 BR;

(a)FR-BR∪BR-FR　//获取有歧义的分词

(c)运用规则处理(2)中的歧义部分

(d)输出 LR , LR 为最后的分词结果.

例如: Text= "组织化解危机. "

利用拼音首字母正向匹配算法, 得到结果 FR={组织化, 解, 危机}.

利用拼音首字母逆向匹配算法, 得到的结果 BR={组织, 化解, 危机}.

FR-BR∪BR-FR={组织, 组织化, 解, 化解}, 其中交集字串为"化". 根据规则"化"与后继构成动词"化解"且前驱为名词, 切分为"组织"和"化解"两个词, 即 LR={组织, 化解, 危机}.

5 实验分析

对 1998 年人民日报标注语料库中的语句进行分词, 得到近 13.6 万个不同词性的词, 从中抽取 12000 个词构建普通的无词典结构的词库和按照 PFLS 算法的词典结构建词库. 分别用最大正向匹配算法和最大逆向匹配算法调用普通的无词典结构的词库进行分词. 然后与 FPLS 算法采用词典结构进行分词的结果进行对比.

实验结果表明, 拼音首字母自动分词算法时间复杂度比传统的最大正向匹配算法和最大逆向匹配算法相比, 效率高、正确率高. 实验统计结果如表 2 所示.

表 2　实验结果统计

	词语总数(个)	正确数(个)	正确率(%)	平均单位时间
最大正向匹配	3600	3243	90.08	43
最大逆向匹配	3600	3288	91.33	39
FPLS 分词算法	3600	3317	92.13	36

为了更好的说明问题, 本文通过两个例子来论证 FPLS 的可行性, 并对实验过程中出现的错误进行分析.

例 3.“通过组织化解了他们的矛盾”

正向最大匹配的结果为“通过\组织化\解\了\他们\的\矛盾”, 逆向最大匹配算法的结果为“通过\组织\化解\了\他们\的\矛盾”. 两者匹配的结果不一致出现歧义字段“组织化解”, 本文采用规则解决了歧义问题提高了准确率.

例 4.“这道题学生会”

正向最大匹配的结果为“这\道\题\学生会”, 逆向最大匹配算法的结果为“这\道\题\学生会”. 两者虽匹配结果一致, 但是不符合语义. 正确结果是“这\道\题\学生\会”. 出现这种情况的原因是类似这样的语句需要借助语义、语境信息解决. 然而, 并未有一种很好的方法解决语义上歧义的问题, 包括目前较为成熟的分词系统 LTP 也没有很好的解决方法, 这也是分词结果出现错误的原因. 解决语义歧义问题也是我们下一步要做的工作.

基于 FPLS 分词算法的时间之所比最大正向匹配和最大逆向匹配算法短, 是因为 FPLS 算法采用的词典结构采用了多维索引, 而最大正向匹配分词算法和最大逆向分词算法未采用索引.

6 结语

FPLS算法将多维索引与歧义处理规则相结合, 分词效率高且正确率较高, 适用于搜索引擎和快速分词. 由于歧义处理规则针对正向匹配与逆向匹配中的歧义制定, 规则不够全面, 与专门的语义、语法分词歧义处理相比还有一定的差距. 在保证分词的高效的同时, 最大限度提高分词的准确率是进一步研究的课题.

参考文献

1 王瑞雷,栾静,潘晓花,等.一种改进的中文分词正向最大匹配算法.计算机应用与软件,2011,28(3):195–197.

2 周俊,郑中华,张炜.基于改进最大匹配算法的中文分词粗分方法.计算机工程与应用,2014,(2):124–128.

3 张科.多次 Hash 快速分词算法.计算机工程与设计,2007,28(7):1716–1718.

4 李振星,徐泽平,唐卫清,等.全二分最大匹配快速分词算法.计算机工程与应用,2002,38(11):106–109.

5 罗洋.一种基于双哈希二叉树的中文分词词典机制.计算机应用与软件,2013,30(5):251–253.

6 张江.基于规则的分词方法.计算机与现代化,2005,(4):18–20.

7 翟凤文,赫枫龄,左万利.字典与统计相结合的中文分词方法.小型微型计算机系统,2006,27(9):1766–1771.

8 韩冬煦,常宝宝.中文分词模型的领域适应性方法.计算机学报,2015,38(2):272–281.

9 王艳,元昌安,覃晓,等.基于 VC++/MFC 的中文自动分词算法及其软件的实现.广西师范学院学报(自然科学版),2008,25(3):104–108.

10 熊回香.全文检索中的汉语自动分词及其歧义处理.中国图书馆学报,2005,31(5):54–57.

11 赵伟,戴新宇,尹存燕,等.一种规则与统计相结合的汉语分词方法.计算机应用研究,2004,21(3):23–25.