基于Hash结构词典的双向最大匹配分词法

陈之彦[1] 李晓杰[1] 朱淑华[2] 付丹龙[2] 邢诒海[3]
(暨南大学国际学院 广州 510632)[1] (暨南大学信息科学技术学院 广州 510632)[2]
(广州市经济贸易信息中心 广州 510032)[3]

摘 要 针对当前自然语言处理中中文分词基于词典的机械分词方法,正序词典不能作为逆向最大匹配分词词典以及反序词典维护困难的问题,提出一种新的词典构造方法并设计了相应的双向最大匹配算法,同时在算法中加入了互信息歧义处理模块来处理分词中出现的交集型歧义。该算法可以在分词的过程中显著提高分词的精确度,适用于对词语切分精度要求较高的中文语言处理系统。

关键词 分词词典,双向最大匹配法,基于Hash的单字索引,互信息歧义处理
中图法分类号 TP391.1 **文献标识码** A

Bi-direction Maximum Matching Method Based on Hash Structural Dictionary

CHEN Zhi-yan[1] LI Xiao-jie[1] ZHU Shu-hua[2] FU Dan-long[2] XING Yi-hai[3]
(International School, Jinan University, Guangzhou 510632, China)[1]
(School of Information Science and Technology, Jinan University, Guangzhou 510632, China)[2]
(Guangzhou City Economic and Trade Information Center, Guangzhou 510632, China)[3]

Abstract In the Chinese natural language processing, aimming at the problem that ordinary dictionary cannot be used for reverse maximum matching method and it is difficult to maintain a reverse dictionary, we put forward a new kind of dictionary structure and corresponding bi-direction maximum matching method, and added mutual information ambiguity processing block in the algorithm. Compared with the previous maximum matching method, this algorithm can increase the segmentation accuracy significantly. It is applicable to some Chinese natural language processing systems which have high segmentation accuracy requirement.

Keywords Segmentation dictionary, Bi-direction maximum matching method, Single word index based on Hash structure, Mutual information ambiguity processing

1 引言

自然语言处理,是用计算机对中文进行转换、传输、存贮、分析、加工、利用的技术。

在汉语中,词是最小的可独立活动的语义单位,是自然语言处理系统中重要的知识载体和基本的操作单元[1]。中文分词,则是将一个汉字序列切分成一个一个单独的词,是将连续的字序列按照一定的规范重新组合成词序列的过程。在大数据的处理中,中文分词是中文数据处理的基础,一个好的中文分词系统能够有效提高系统在数据处理时对数据的理解能力,增强对结构化数据的提取能力,因此一个优秀的中文分词算法能有效提升用户对分词系统乃至数据处理系统的评价。然而在中文文本中,词与词之间并没有像英文一样以空格作为自然分界符,词语的边界较难区分,因此中文的自然语言处理的基础是对要处理的文本进行精准、快速的分词。

2 中文分词法现状

在现有的中文分词技术中所使用的分词方法大体可以分为3类:基于理解的分词方法、基于统计的分词方法以及基于词典的分词方法。

基于理解的分词方法不仅要求要有很好的分词词典,而且还需要加入语义和句法的分词。通过获取相关词和句子的语义信息来对分词产生的歧义进行判断,从而模拟人类对句子的理解过程。但由于汉语语言知识的笼统性和复杂性,很难将各种信息转化成机器可直接读取的形式。

基于统计的分词方法主要靠一个或多个具有代表性的规模相对较小的训练语料库获得相关信息统计的数据,再根据语料库中得到的数据来指导分词的进行。而其中比较重要的

本文受国家自然科学基金(61272415,61272067),国家863计划重大项目(2013AA01A212),广东省自然基金团队研究项目(S2012030006242),广州市重点实验室开放基金(2012-224)资助。

陈之彦 男,主要研究方向为大数据与信息安全,E-mail:657287638@qq.com;**李晓杰**(1993—),男,主要研究方向为大数据与信息安全,E-mail:447668801@qq.com;**朱淑华** 女,博士,硕士生导师,主要研究方向为计算机网络、云计算安全、大数据,E-mail:zsh@jnu.edu.cn(通信作者);**付丹龙** 男,硕士,主要研究方向为大数据、云计算,E-mail:1187747241@qq.com;**邢诒海** 男,高级工程师,主要研究方向为大数据、云计算,E-mail:297053188@qq.com。

方法有基于最大概率的分词法、基于层叠隐马尔科夫模型和条件随机场模型的分词法，这些方法在通用分词领域里取得了不错的分词效果，但由于依赖统计算法，需要有大量的语料库作为分词的基础。当处理文本量较大时，词频的计算需要耗费大量的时间，加之语料库不同的训练方式会导致分词的结果产生很大的不同，该种方法的分词效率并不是很高[2,3]。

基于词典的分词方法也可称为基于字符串匹配的分词方法，是根据分词词典和一个基本的切分规则来进行词语的切分，由于分词规则简单、切分速度较快而得到广泛应用。分词的方法大致可分为正向匹配、逆向匹配与双向匹配3种，按照是否与词性标注的过程相结合，又可分为分词与词性标注一体化和单纯的分词方法。

在基于词典(字符串匹配)的分词方法的研究中，主要的研究方向大多都是对正向最大匹配分词算法进行改进，而主要的改进有对词典的结构进行改进、动态确定正向最大匹配中的最大词长、减少无用的匹配次数，提高了词典中的词条匹配速度[4-6]。与这些改进的方案相比，本文所提出的双向最大匹配算法所比较的单词最大词长是由单词的首字或末字动态决定的，不会出现无用的单词匹配。

而由于逆向最大匹配分词算法中的反序词典结构要求以词语的最后一个字来建立索引，而且待比较字段都要逆序以适应反序词典，如词语“临危不惧”在词典中存放为“惧不危临”，并不符合人们正常的思维，词典的维护相当困难，而在字段的匹配中还要再进行一次反序排列，从而会对效率产生影响[7]。但一般来说，逆向匹配分词算法的切分精度略高于正向匹配分词算法。统计结果表明，单纯使用正向最大匹配分词算法的切分错误率约为1/169，单纯使用逆向最大匹配分词算法的切分错误率约为1/245[8]。而传统的双向最大匹配分词法，则是对语料先后进行正向最大匹配分词法和逆向最大匹配分词法，再按某种规则对不同的切分结果进行选择，能够在一定程度上提升正向最大匹配分词法和逆向最大匹配分词法的分词精度。但由于需要对语料进行二次分词，因此双向最大匹配分词法在速度上难与单向最大匹配分词法相比较。

在基于词典的分词方法中，有两大难题一直难以解决，一是分词中的歧义识别问题，而分词中的歧义主要有两种：组合型歧义和交集型歧义，在字串AB中，如果A，B和AB都是一个词，则AB存在组合型歧义，如“风雨”，“风”和“雨”都是词，但是它们合起来也是一个词；而交集型歧义就是指字串ABC中，AB和BC都是一个词，两个相邻的词之间有重叠的部分，如“进行了有关实验”，“进行”是一个词，“行了”也是一个词，它们重用了一个“行”字。研究表明，歧义的产生主要是交集型歧义，约占整个分词歧义的90%[9]。第二个难题则是新词的识别问题，如人名、地名、产品名、简称、专业术语、省略词等等，这些词没有被收录在词典中，但又确实能称为词。计算机在识别新词方面仍存在非常多的困难，但这些新词也是人们在日常生活中经常使用的词，因此对于分词系统来说，对新词的识别也是十分重要的。

鉴于此，本文基于传统中文分词使用的正向最大匹配法与逆向最大匹配法，设计了一种双向最大匹配分词法与对应的词典结构，并在算法中加入了互信息处理模块以提高算法对分词中歧义的识别能力，明显地提高分词的精度与准确度。对于新词的处理，本文为该种词典提供了可靠的维护方法。

3 分词词典结构设计

本文设计的分词词典主要分为两个部分：单字索引表和词典正文。在对语料进行切分匹配时，算法以要开始进行匹配的字作为索引项，利用单字索引表得到与该字相关的词语的开始位置，然后按照顺序进行词语的匹配切分。

在单字索引表中每个单元包括3项，如图1所示：①单字：要进行检索的字。因为分词算法为双向匹配，所以要进行检索的字可能为词语的第一个字也可能为词语的最后一个字。②第一项指针：该指针指向以检索字为首字的第一个词语(即以检索字为首字的长度最短的词语)。③第二项指针：该指针指向以检索字为最后一个字的第一个词语(即以检索字为最后一个字的长度最短的词语)。

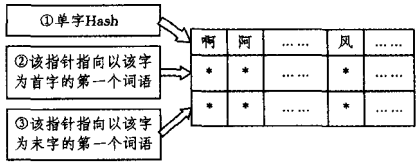

图1 单字索引表结构

在该词典结构中，词典被拆分成单个的词语来存放，如图2所示，每个词语的结构都包括4项：①词语：要进行匹配的词语。②第一项指针：该指针指向与该词语首字相同的下一个词语，若没有则为空。③第二项指针：该指针指向与该词语最后一字相同的下一个词语，若没有则为空。④长度：该词语的长度。

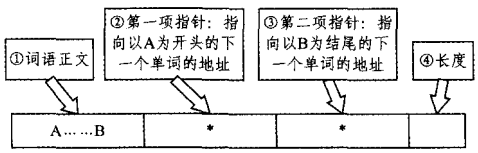

图2 词语索引表结构

词语在该词典结构中的存放是按照词语的长度来排序的，起始的指针从单字索引表开始，若以某个单字A为首字的最大词长为N，则单字索引表中该字的第一项指针指向以A为首字的长度最短的词语，而该词语的第一项指针指向以A为首字的长度次短的词语，以此类推，最后指针会指向以A为首字的词长为N的词语，而该词语的第一项指针为NULL，如图3所示。

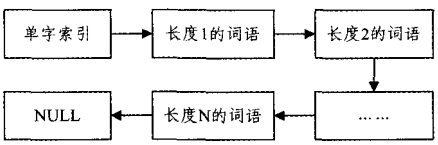

图3 指针方向示意图

最后的总体结构如图4所示。

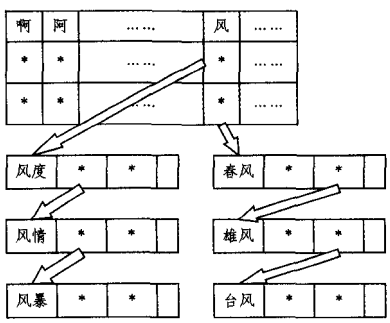

图4 总体结构

4 双向最大分词算法设计

整个双向最大匹配分词算法分为3个模块：①正向匹配模块；②逆向匹配模块；③互信息歧义处理模块。

4.1 正向最大匹配分词处理模块

正向最大匹配分词处理模块中，每次匹配前都要对待切分字符串长度进行检测，若长度小于或等于1，则该模块切分结束。当待切分字符串进入该模块时，先在单字检索表中检索首字，然后取出第一项指针中的词，对待切分字符串中相同长度的字段进行匹配。匹配成功则添加切分标记，匹配失败则继续取出第一项指针中的词进行匹配。重复该步骤，直至第一项指针为NULL。然后按切分标记对字符串进行切分。重复“匹配-切分”的步骤，直至待切分字符串长度小于或等于1。

正向最大匹配分词处理模块如图5所示。

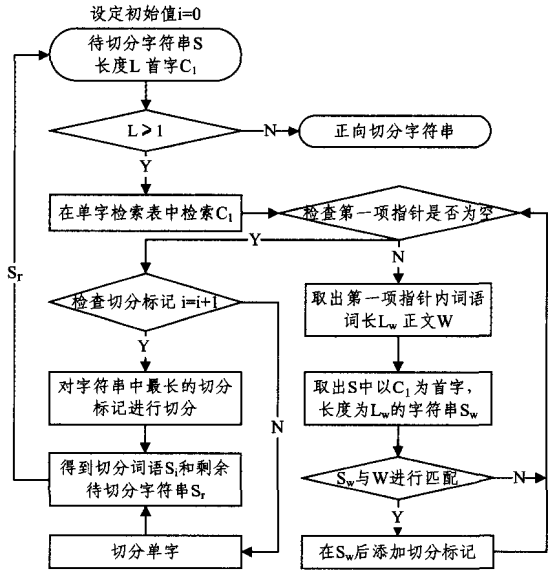

图5 正向最大匹配分词处理模块

正向最大匹配分词处理模块的执行步骤如下：

(0)设定初始值 i 用于切分计数，令 $i=0$。

(1)对待切分字符串 S 的长度 L 进行判定，若 L 小于1，则该模块切分结束；若 L 大于或等于1，则跳转至(2)。

(2)在单字索引表中检索 S 的首字 C_1，跳转至(3)。

(3)检查第一项指针是否为空，若第一项指针为空，则跳转至(7)；若不为空，则跳转至(4)。

(4)取出指针中的词语，得到词语长度 L_w 与词语正文 W，跳转至(5)。

(5)若词语长度 L_w 大于待切分字符串长度 L，则跳转至(7)；若否，跳转至(6)。

(6)取出 S 中以 C_1 为首字、长度为 L_w 的字符串 S_w，对 S_w 与 W 进行匹配。若匹配成功，则添加切分标记，跳转至(3)。若匹配不成功，则直接跳转至(3)。

(7)检查字符串中是否有切分标记，并使 $i=i+1$。若有，则按最长的切分标记切分句子；若没有切分标记，则切分单字。得到切分词语 S_i 和剩余待切分字符串 S_r，将 S_r 作为新的 S，跳转至(1)。

4.2 逆向最大匹配分词处理模块

与正向最大匹配分词处理模块相似，在匹配前要对待切分字符串长度进行检测，若长度小于或等于1，则该模块切分结束。当待切分字符串进入该模块后，先在单字检索表中检索末字，然后取出第二项指针中的词，从待切分字符串的尾部开始，对待切分字符串中相同长度的字段进行匹配。匹配成功则添加切分标记，匹配失败则继续取出第一项指针中的词进行匹配。重复该步骤，直至第二项指针为NULL。然后按切分标记对字符串进行切分。重复“匹配-切分”的步骤，直至待切分字符串长度小于或等于1。

逆向最大匹配分词处理模块如图6所示。

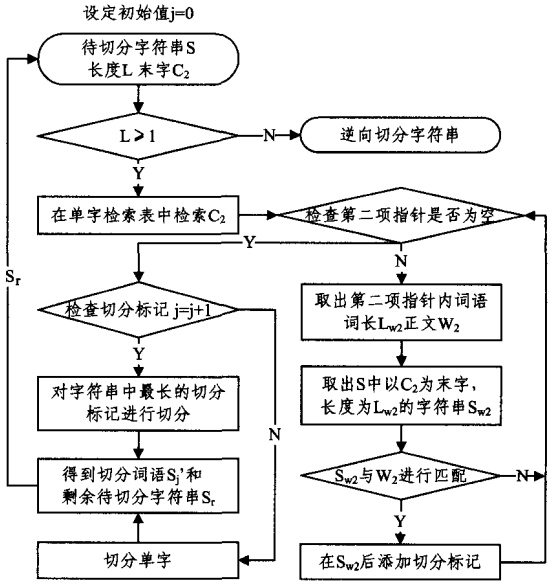

图6 逆向最大匹配分词处理模块

逆向最大匹配分词处理模块的执行步骤如下：

(0)设定初始值 j 用于切分计数，令 $j=0$。

(1)对待切分字符串 S 的长度 L 进行判定，若 L 小于1，则该模块切分结束；若 L 大于或等于1，则跳转至(2)。

(2)在单字索引表中检索 S 的最后一个字 C_2，跳转至(3)。

(3)检查第二项指针是否为空，若第二项指针为空，则跳转至(7)；若不为空，则跳转至(4)。

(4)取出指针中的词语，得到词语长度 L_{w2} 与词语正文 W_2，跳转至(5)。

(5)若词语长度 L_{w2} 大于待切分字符串长度 L，则跳转至(7)；若否，跳转至(6)。

(6)取出 S 中以 C_2 为最后一个字、长度为 L_{w2} 的字符串 S_{w2}，对 S_{w2} 与 W_2 进行匹配。若匹配成功，则添加切分标记，跳转至(3)；若匹配不成功，则直接跳转至(3)。

(7)检查字符串中是否有切分标记，并使 $j=j+1$。若有，则按最长的切分标记切分句子；若没有切分标记，则切分单字。得到剩余待切分字符串 S_r 和切分词语 S_j'，将 S_r 作为新的 S，跳转至(1)。

4.3 互信息歧义处理模块

4.3.1 传统的最大概率分词算法

为了增强分词系统的信息处理能力，传统的匹配分词算法中大多加入了基于最大概率的分词算法来帮助优化分词结果。

使用最大概率分词算法进行分词的基本思想是一个待切

分的字符串可能有多种切分的结果，而将其中出现概率最大的作为该字符串的分词结果。如：对于待切分字符串 S 可能的切分结果 S_1 与 S_2，最大概率分词算法通过比较 $P(S_1|S)$ 与 $P(S_2|S)$ 值的大小来决定最后采用哪种分词结果。

比较 $P(S_1|S)$ 与 $P(S_2|S)$ 的大小，即比较 $P(S_1)$ 与 $P(S_2)$ 的大小。根据一元语法，词语之间出现的概率互相独立，则有：

$$P(S)=P(S_1,S_2,\cdots,S_n)=P(S_1)P(S_2)\cdots P(S_n)$$

即要求字符串切分结果的概率，就是求构成该切分结果的各个词的概率之积。每个词的概率等于该词在语料中出现的次数除以语料的总词数。

基于最大概率的分词算法的确在一定程度上处理了部分歧义问题，但也存在一定的问题。如果切分的字符串较长，则可能存在多种切分结果，计算量则会大幅度增加，从而影响系统的分词效率。

4.3.2 互信息处理算法

将字串 AB 中 A 和 B 之间的互信息定义为：

$$I(A,B)=\log_2\frac{P(A,B)}{P(A)P(B)}$$

其中，字串 AB 联合出现的概率表示为 $P(A,B)$，词条 A 出现的概率表示为 $P(A)$，词条 B 出现的概率表示为 $P(B)$，它们在文本中出现的次数分别计为 $n(AB)$，$n(A)$，$n(B)$，n 为词频总数，则有：

$$P(A,B)=\frac{n(AB)}{n},P(A)=\frac{n(A)}{n},P(B)=\frac{n(B)}{n}$$

互信息反映了两词之间的相关程度，如果 $I(A,B)>0$，那么 A 和 B 是相关的，随着 $I(A,B)$ 值的增加，则 A 和 B 之间的相关度增加。假定 $I(A,B)$ 大于某个给定的值，则可以认为 AB 组合成词[10]。通过对比歧义字段中单字之间的互信息度，可以在一定程度上对交集型歧义进行处理。

相比于最大概率分词算法，基于互信息的歧义处理算法不需要对整个待切分字符串的概率进行计算，系统只需要比较歧义字段中的词语的互信息，计算量大大减小。互信息歧义处理模块如图 7 所示。

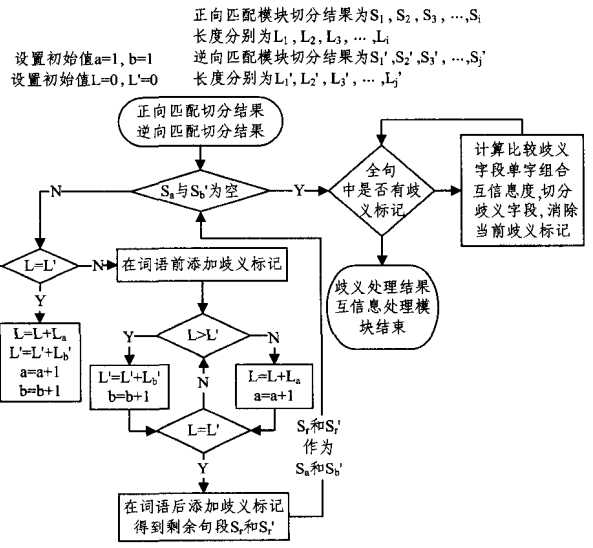

图 7 互信息歧义处理模块

互信息歧义处理模块的执行步骤如下：

(0)正向匹配模块切分结果为 $S_1,S_2,S_3,\cdots,S_i$，长度分别为 $L_1,L_2,L_3,\cdots,L_i$；逆向匹配模块切分结果为 $S_1',S_2',S_3',\cdots,S_j'$，长度分别为 $L_1',L_2',L_3',\cdots,L_j'$

设置初始值 a 和 b 用于计数，令 $a=1,b=1$。

设置初始值 L 用于表示正向匹配切分片段长度，令 $L=0$。

设置初始值 L' 用于表示逆向匹配切分片段长度，令 $L'=0$。

(1)检查 S_a 和 S_b' 是否为空，若 S_a 与 S_b' 都为空，则跳转至(5)；若 S_a 与 S_b' 不同时为空，则跳转至(2)。

(2)比较 L 和 L'，若 $L=L'$，则令 $L=L+L_a$，$L'=L'+L_b'$，使 $a=a+1$，$b=b+1$，继续进行(2)；若否，则当前词语为歧义字段的开始，在词语前添加歧义标记，跳转至(3)。

(3)若 $L>L'$，则令 $L'=L'+L_b'$，使 $b=b+1$，跳转至(4)；若 $L<L'$，则令 $L=L+L_a$，使 $a=a+1$，跳转至(4)。

(4)比较 L 和 L'，若 $L\neq L'$，则跳转至(3)；若 $L=L'$，则当前词语为歧义字段的结束，在词语后添加歧义标记，得到剩余句段 S_r 与 S_r'，将 S_r 与 S_r' 作为新的 S_a 与 S_b'，跳转至(1)。

(5)检查句子中是否有歧义标记，若没有歧义标记，则互信息歧义处理模块结束；若有歧义标记，则计算并比较歧义字段单字组合的互信息度，切分歧义字段，消除当前歧义标记，继续跳转至(5)。

5 词典的维护

在网络如此发达的今天，每年都有大量的新词产生并被收入汉语词典，而分词词典也需要做到与时俱进，及时收录新词。而词典的维护则需要实现对词典新词的添加与对错误词的删除。本文中所使用的词典结构能通过对指针的移动来实现对新词的添加与对错误词的删除。

5.1 词语的添加

首先在词典中查询是否有与需要添加的新词重复的词语。若有重复的词语，则取消添加新词；若没有重复的词语，则分两步添加新词，先把新词插入相同首字与相同词长的词语的最末端，确定新词的第一项指针，然后把新词插入相同末字与相同词长的词语的最前端，确定新词的第二项指针。如图 8 所示。

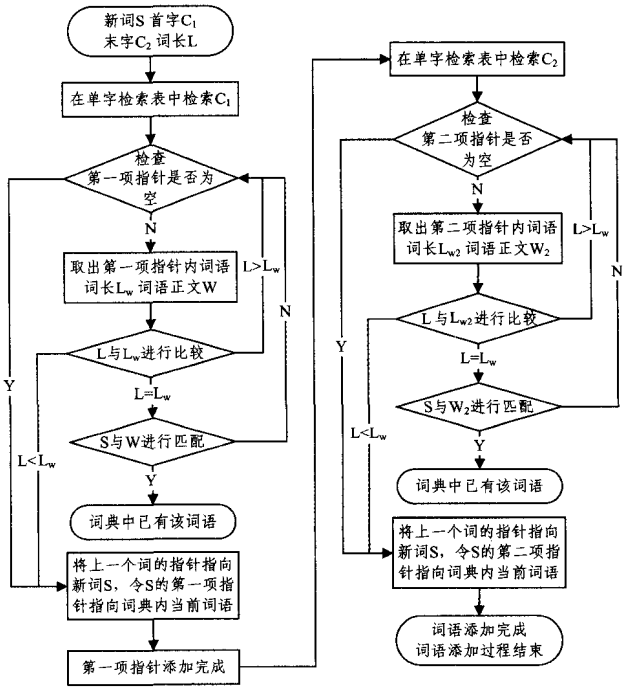

图 8 词语添加过程

(0)先得到新词 S 的首字 C_1、最后一个字 C_2 与词长 L。

(1)在单字检索表中检索 C_1，跳转至(2)。

(2)检查第一项指针是否为空。若不为空，取出第一项指针内的词语，得到词长 L_w 与词典正文 W，跳转至(3)；若第一项指针为空，则跳转至(5)。

(3)进行词长的匹配，若 $L>L_w$，跳转至(2)；若 $L=L_w$，跳转至(4)；若 $L<L_w$，跳转至(5)。

(4)进行新词 S 与词语 W 的匹配，若匹配成功，则输出词典中已有该词语，结束词语添加；若匹配失败，则跳转至(2)。

(5)改变上一个词的指针，令其指向新词 S，令 S 的第一项指针指向词典内当前词语，则第一项指针添加完成。开始第二项指针的添加，跳转至(6)。

(6)在单字检索表中检索 C_2，跳转至(7)。

(7)检查第二项指针是否为空。若不为空，取出第二项指针内的词语，得到词长 L_{w2} 词典正文 W_2，跳转至(8)；若第二项指针为空，则跳转至(10)。

(8)进行词长的匹配，若 $L>L_{w2}$，跳转至(6)；若 $L=L_{w2}$，跳转至(9)；若 $L<L_{w2}$，跳转至(10)。

(9)进行新词 S 与词语 W_2 的匹配，若匹配成功，则输出词典中已有该词语，结束词语添加；若匹配失败，则跳转至(6)。

(10)改变上一个词的指针，令其指向新词 S，令 S 的第二项指针指向词典内当前词语，则第二项指针添加完成。输出词语添加完成，词语添加过程结束。

词语的添加过程示意图如图 9 所示。

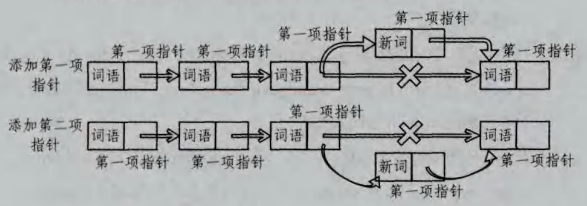

图 9　词语添加示意图

5.2　词语的删除

与词语的添加相同，首先查询需要删除的词语是否在词典中。若不存在于词典中，则取消删除词语；若存在于词典中，也是分两步对词语进行删除。过程与对词语的添加相似，词语的删除过程如图 10 所示。

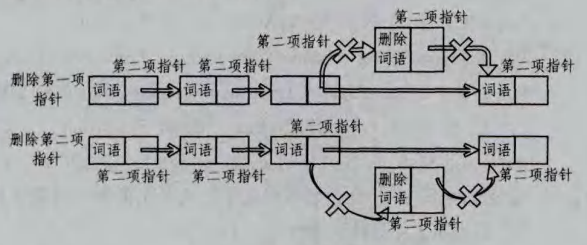

图 10　词语删除示意图

6　算法的实现与分析

6.1　双向最大匹配分词系统的实现

系统环境：Window7 64 位操作系统

实验环境：Eclipse

CPU：Intel(R) Xeon(R) CPU E3-1230 v3 @ 3.30GHz

内存：Kingston DDR3 1600MHz 8GB

硬盘：Seagate ST1000DM003 1TB

基于词典的双向最大匹配分词系统如图 11 所示，可以手动输入需要分词的语料或者载入现有的语料文件进行分词。用户也可以选择载入其他类型不同的字典以提高分词的精确度。

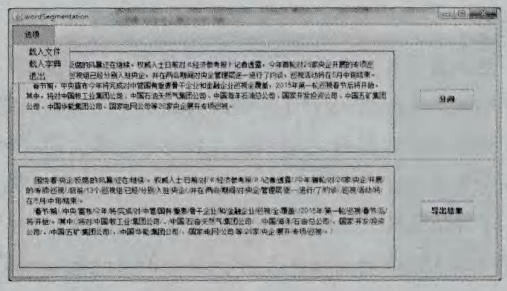

图 11　基于词典的双向最大匹配分词系统

在输入语料后点击分词，系统会同步对语料进行正向最大匹配分词与逆向最大匹配分词，然后在互信息歧义处理模块中处理歧义语段，分词后得到的结果会显示在输出框内，用户可以选择导出结果，将分词结果保存，以便进行后续的语料处理。

选取 1998 年 1 月《人民日报》语料库作为测试用语料，文件总大小为 8.42MB。为了处理方便，将该文件分成 3 个子文件进行该分词系统的测试。

6.2　数据分析

对分词系统的评判标准为分词的准确率与召回率。准确率为在分词结果中，切分正确的词语数与分词系统切分的词语数的比值。召回率为在分词结果中切分正确的词语数与切分语料中总的正确词语数的比值。

应用本文设计的基于词典的双向最大匹配分词系统对测试语料进行分词，与传统的双向最大匹配法分词和基于最大概率的双向最大匹配分词法进行对比，得到了不同的分词结果，如表 1 所列。

表 1　不同语料分词统计结果

方法	文件名	文件大小(MB)	准确率(%)	召回率(%)	分词速度(/s)
互信息双向最大匹配分词法	199801_01	3.00	95.15	94.82	24.25
	199801_02	3.00	95.80	95.11	24.91
	199801_03	2.42	95.57	95.42	22.38
传统的双向最大匹配分词法	199801_01	3.00	92.14	91.78	56.74
	199801_02	3.00	91.80	91.20	52.46
	199801_03	2.42	93.47	93.05	45.82
最大概率双向最大匹配分词法	199801_01	3.00	95.27	95.03	235.25
	199801_02	3.00	95.18	94.98	233.96
	199801_03	2.42	94.43	94.03	210.47

对语料进行了切分，其中包括：本文提出的改进的双向最大匹配法，传统的正向最大匹配法，传统的逆向最大匹配法，传统的双向最大匹配法和最大概率双向最大匹配法。使用各方法对语料切分的准确率与召回率的统计结果如表 2 所列。

表 2　分词精度统计结果

分词方法	准确率(%)	召回率(%)
互信息双向最大匹配法	95.50	95.10
正向最大匹配法	90.65	90.53
逆向最大匹配法	91.17	90.98
传统的双向最大匹配法	92.40	91.94
最大概率双向最大匹配法	94.95	94.72

由表1与表2可以看出，对于语料的切分，本文中所提供的词典结构以及双向最大匹配法，与传统的单向最大匹配分词法和双向最大匹配分词法相比较，能够显著提升分词的精度与分词的速度；而与基于最大概率的双向最大匹配法相比，分词的准确率与召回率相差无几，却大大提升了分词的速度。

结束语 本文基于传统中文分词使用的正向最大匹配法与逆向最大匹配法，设计了一种既可用于正向最大匹配也可用于逆向最大匹配的词典结构，并设计与之相应的双向最大匹配分词法，提升了传统双向最大匹配分词的速度，并在最后加入了互信息歧义处理，进一步提升了分词的精度。相比较于基于最大概率的分词算法，本文提出的双向最大匹配分词算法在速度上有优势。因而本文提出的基于词典的双向最大匹配分词算法可以用于对词语分词精度要求更高的中文语言处理系统中，如车载人机交互的语音识别系统以及非结构化-结构化信息储存系统或非结构化大数据信息处理系统，以提高系统对信息的识别能力、预测分析能力，提升系统分析预测的准确度。

参 考 文 献

[1] 奉国和，郑伟. 国内中文自动分词技术研究综述[J]. 图书情报工作，2011，55(2)：41-45

[2] 罗智勇，宋柔. 现代汉语通用分词系统中歧义切分的实用技术[J]. 计算机研究与发展，2006，43(6)：1122-1128

[3] 吴育良. 百度中文分词技术浅析[J]. 河南图书馆学刊，2008(8)：115-117

[4] 莫建文，郑阳，首照宇，等. 改进的基于词典的中文分词方法[J]. 计算机工程与设计，2013，34(5)：1802-1807

[5] 吴旭东. 正向最大匹配分词算法的分析与改进[J]. 科技传播，2011(20)

[6] 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[J]. 计算机应用与软件，2011，28(3)：195-197

[7] 张李义，李亚子. 基于反序词典的中文逆向最大匹配分词系统设计[J]. 现代图书情报技术，2006(8)：42-45

[8] 赵艳红，费洪晓. 一个基于改进的反序分词词典的中文分词算法[J]. 深圳职业技术学院学报，2004，3(4)：28-31

[9] 罗桂琼，费洪晓，戴弋. 基于反序词典的中文分词技术研究[J]. 计算机技术与发展，2008，18(1)：80-83

[10] 丁振国，张卓，黎靖. 基于 Hash 结构的逆向最大匹配分词算法的改进[J]. 计算机工程与设计，2008，29(12)：3208-3211

(上接第 41 页)

参 考 文 献

[1] 荣辉桂，火生旭，胡春华，等. 基于用户相似度的协同过滤推荐算法[J]. 通信学报，2014，35(2)：16-24

[2] Yin Pei-feng, Luo Ping, Wang C-L, et al. Silence is Also Evidence: Interpreting Dwell Time for Recommendation from Psychological Perspective[C]// the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), 2013. Chicago: ACM Press, 2013: 989-997

[3] 李建廷，郭晔，汤志军，等. 基于用户浏览行为分析的用户兴趣度计算[J]. 计算机工程与设计，2012，33(3)：968-972

[4] Ricci F, Rokach L, Shapira B. Recommender Systems Handbook [M]. New York: Springer, 2010

[5] Rendle S, Freudenthaler C, Gantner Z, et al. BPR: Bayesian personalized ranking from implicit feedback[C]// 25th Conference on Uncertainty in Artificial Intelligence, 2009. Virginia: AUAI Press, 2009: 452-461

[6] Yi Xing, Hong Liang-jie, Zhong Er-heng, et al. Beyond Clicks: Dwell Time for Personalization[C]// 8th ACM Recommender Systems Conference, 2014. Foster City: ACM Press, 2014: 113-120

[7] Gong Song-jie. A Collaborative Filtering Recommendation Algorithm Based on User Clustering and Item Clustering[J]. Journal of Software, 2010, 5(7): 745-752

[8] Gong Song-jie, Cheng Guang-hua. Mining User Interest Change for Improving Collaborative Filtering[C]// 2nd International Symposium on Intelligent Information Technology Application, 2008. Shanghai: IEEE, 2008: 24-27

[9] Yu Li, Liu Lu, Li Xue-feng. A Hybrid Collaborative Filtering Method for Multiple-interests and Multiple-content recommendation in E-Commerce[J]. Expert Systems with Applications, 2005, 28: 67-77

[10] Billsus D, Pazzani M J. Learning Collaborative Information Filters[C]// 15th International Conference on Machine Learning, 1998. Madison: ICML, 1998: 46-54

[11] Funk S. Netflix Update: Try This at Home[EB/OL]. http://sifter.org/~simon/journal/20061211.html. 2006 December

[12] 孙光福，吴乐，刘淇，等. 基于时序行为的协同过滤推荐算法[J]. 软件学报，2013，24(11)：2721-2733

[13] Koren Y. Collaborative filtering with temporal dynamics[C]// 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2009. Paris: ACM Press, 2009: 89-97

[14] Claypool M, Le E, Waseda M, et al. Implicit interest indicators [C]// The 6th International Conference on Inteligent User Interfaces (IUI), 2001. New York: ACM Press, 2011: 33-40

[15] Wu Xiao-jun, Feng Ai-gang, Yin Qin-ye. Universal Discrete Model and Linear Algebra Representation for Variant OFDM-CDMA Systems[C]// IEEE International Symposium on Circuits and Systems, 2001. Sydney: IEEE, 2001: 213-216

[16] 付关友，朱征宇. 个性化服务中基于行为分析的用户兴趣建模[J]. 计算机工程与科学，2005，27(12)：76-78

[17] Sarwar B, Karypis G, Konstan J et al. Item-Based collaborative filtering recommendation algorithms[C]// 10th International World Wide Web Conference, 2001. HongKong: ACM Press, 2001: 285-595