基于 Delphi 的中文分词设计①

Chinese Split Word Design Based on Delphi

刘建培　(广东商学院 教育技术中心 广东 广州 510320)

摘　要： 词是语言中最小的能够独立活动的有意义的语言成分,是信息处理的基本单位。词频统计、语音识别、字符识别、自动分类、机器翻译、信息检索、信息抽取等方面的研究， 也必须首先分词。在这些应用和研究领域，没有准确高效的分词策略，汉语的进一步分析必将受到严重影响。本文研究用 Delphi 设计中文分词。

关键词： 中文分词 Delphi 词频 正向最大匹配

1　引言

汉语自动分词是对汉语文本进行自动分析的第一个步骤。可以这样设想汉语自动分词过程的困难：如果把某个英语文本中的所有空格符都去掉，然后让计算机自动恢复文本中原有的空格符，这就是词的识别过程。分词体现了汉语与英语的显著的不同。英语文本是小字符集上的已分隔开的词串，而汉语文本是大字符集上的连续字串。把字串分隔成词串，就是自动分词系统需要做的工作。

汉语分词又是各种中文信息处理应用系统中共同的、基础性的工作，例如：语音识别、字符识别、语音合成、文本校对、文本摘要、词频统计、作家作品风格学研究、自动标引、自动分类、信息检索和机器翻译、信息抽取等。在这些应用和研究领域， 没有准确高效的分词策略，汉语的进一步分析必将受到严重影响。

词是最小的能独立活动的有意义的语言成分。计算机的所有语言知识都来自机器词典(给出词的各项信息)、句法规则(以词类的各种组合方式来描述词的聚合现象)以及有关词和句子的语义、语境、语用知识库。汉语信息处理系统只要涉及句法、语义就需要以单词作为基本单位，例如汉字的拼音——字转换、简体——繁体转换、汉字的印刷体或手写体的识别、汉语文章的自动朗读(即语音合成)等等，都需要使用词的信息。分词以后在词的层面上做转换或识别，处理的确定性就大大提高了。再如信息检索，如果不分词(按字检索)，当检索德国货币单位“马克”时，就会把“马克思”检索出来，而检索“华人”时会把“中华人民共和国”检索出来。如果进行分词，就会大大提高检索的准确率。在更高一级的文本处理中， 例如句法分析、 语句理解、自动文摘、自动分类和机器翻译等，更是少不了词的详细信息。

2　中文分词的词库设计

计算机上使用的汉字有两类代码，一类叫外码，用来输入汉字，如拼音码、五笔字型码等。在同一计算机中，可以存在多种外码，根据需要选择汉字输入码。另一种叫内码，同一计算机中只有一种内码，用不同外码输入的汉字，都要转换成相同的内码存储到计算机中。汉字内码有 GB2312、BIG5、GBK,GB2312 是内码的一种，在同一计算机中内码只能有一种，但根据需要可以设置为不同的内码。区位码是一个四位的十进制数，每个区位码都对应着一个唯一的汉字或符号，它的前两位叫做区码，后两位叫做位码。一个汉字的内码由两个字节组成。汉字内码与区位码之间有一个简单的数学关系：

内码第一字节＝区码+160

内码第二字节＝位码+160

比如，查区位码表知道，“啊”字在 16 区 01 位，它的内码为：

① 收稿时间:2008-09-10

第一字节=16+160=176；

第二字节=1+160=161。

分词效率很大程度取决词库的设计，词库设计必须实现最大限度的查找匹配词。词库设计主要是两大部分，词数据部分和索引部分。词数据部分将词按照首字第一字节的内码从小到大排列，首字相同的词按照词频从大到小排列。索引部分包含词索引，词索引是针对词首字的索引，索引记录的是首字第一字节的内码。在具体使用该词库时，需要将整个词库装到内存，按首字第一字节内码提供索引。这个词库优点是查询速度很快，数量越大的词库则速度比纯文本词库就越快。

为了得到高效合适的词库，采用搜狗实验室 http://ww w.sogou.com/labs/的互联网词库，搜狗拼音输入法就是用该词库,该词库来自于对 SOGOU 搜索引擎所索引到的中文互联网语料的统计分析，统计所进行的时间是 2006 年 10 月,涉及到的互联网语料规模在 1 亿页面以上。统计出的词条数约为 15 万条高频词，本人按词条首字第一字节内码进行分组，同一组按词频从大到小排列。由于词库过大,必须借助 excel 进行处理。按内码分组的 Delphi 关键代码如下：

```
begin
ExcelApp:=CreateOleObject( 'Excel.Application' );
ExcelApp.WorkBooks.Open(ExtractFileDir(paramstr(0))+'\ck.xls');
//excelapp 类型为 variant 的变量，ck.xls 为词库文件
ExcelApp.Visible := True;
ExcelApp.WorkSheets[1].Activate;
//以上代码用 excel 打开词库文件
K2:=3; //按内码分组输出的列号
for i:=176 to 255 do //汉字内码范围
begin
k:=1; //按内码分组输出的行号
for j:=1 to 65530 do //从 1 行至 65530 行
begin
tmp:=ExcelApp.Cells[j,1].Value; //读取未分组单元格词组
if tmp<>'' then
if ord(tmp[1])=i then //比较首字第一字节的内码
begin
excelapp.cells[k1,k2].value:=tmp; //存放分组结果
k:=k+1; //行递增
end;
end;
k2:=k2+1; //列递增
end;
if not ExcelApp.ActiveWorkBook.Saved then
    ExcelApp.ActiveSheet.PrintPreview;
//excel 保存操作
end.
```

由于最大匹配法必须首先设定一个匹配词长的初始值，这个长度限制是最大匹配法在效率与词长之间的一种妥协。我们来看一下以下两种情况：

(1) 词长过短，长词就会被切错。例如当词长被设成 5 时，也就意味着它只能分出长度为 5 以下词，例如当这个词为“中华人民共和国”长度为 7 的词时，我们只能取出其中的 5 个字去词库里匹配，例如“中华人民共”，显然词库里是不可能有这样的词存在的。因此我们无法正确的划分出“中华人民共和国”这样的词长大于 5 的词。

(2) 词长过长，效率就比较低。也许有人会认为既然 5 个字无法满足我们的分词要求，何不将词长加大，例如加到 10 或者 100，毕竟这个世界超过 100 个字长的词还是很少见的，我们的词长问题不就解决了？然而当词长过长时，我们却要付出另一方面的代价：效率。效率是分词算法、甚至是整个算法理论体系的关键，毕竟算法书里所有的高深的查询或排序算法都是从效率出发的，否则任何笨办法都可以解决分词效率低的问题。设想到我们把字长设成 100 个词时，我们必须将词从 100 开始一直往下匹配直到找到要查的字为止，而我们大多数词的字长却只有两三个字，这意味着前 97 次的匹配算法是徒劳的。因此我们必须要在词长与效率之间进行妥协，既要求分词尽量准确，又要求我们的词长不能太长，本文按实际设定最大匹配词长为 8 为最合适，因搜狗实验室的互相网词库有词条的词长超过 8，则必须删除这些词长超过 8 的词条，删除长词组的 Delphi 关键代码为：

```
begin
//首先用 excel 打开词库文件，同前面内码分组的代
```

```
码
//删除词长大于 8 的词组
for j:=1 to 33 do  //行号
begin
tmp:=ExcelApp.Cells[j,1].Value; //读取单元格词条
while length(tmp)>8 do //判断词长是否大于 8
begin
ExcelApp.ActiveSheet.Rows[j].Delete; //词长大于 8 则删除
tmp:=ExcelApp.Cells[j,1].Value; //读取下一个词条
end;
end;
end;
```

3 中文分词算法的设计

词如何切分出来是一个很关键的问题。词是最小的、能独立活动的、有意义的语言成分。汉语的处理只要涉及句法、语义(如检索、翻译、文摘、校对等应用)，就需要以词为基本单位。分词以后，在词的层面上进行处理，其确定性就大大提高了。考虑到效率要求，采用了基于字符串匹配的基本分词方法—正向最大匹配(MM)法则。正向最大匹配法切分精度较高，同时，通过配置它还可以实现完全基于字词的匹配。

最大匹配 (MM)法则也叫“字符串匹配”，在孙宾的《现代汉语文本的词语切分技术》中对其定义如下：基于字符串匹配的分词方法，又叫做机械分词方法，它是按照一定的策略将待分析的汉字串与一个“充分大的”机器词典中的词条进行匹配，若在词典中找到某个字符串，则匹配成功(识别出一个词)。按照扫描方向的不同，串匹配分词方法可以分为正向匹配和逆向匹配；按照不同长度优先匹配的情况，可以分为最大(最长)匹配和最小(最短)匹配；按照常用词优先匹配的情况，可以分为最大词频匹配；按照是否与词性标注过程相结合，又可以分为单纯分词方法和分词与标注相结合的一体化方法。

本文采用了正向最大词频的匹配方法(由左到右的方向)来切词，并按词条首字第一字节的内码进行索引，分词步骤如下：

(1) 根据中文的语言规范和特点，先制定一个停用词表，表中存放那些明显不能构成词的单虚字、助词等；

(2) 分词时不处理文档中含有的停用词表中的字、词，以减少不必要的资源浪费，提高分词速度；

(3) 去除停用词后，以句子为单位，按汉字内码索引，将字符串与中文词表中的词按最大词频进行匹配，若在词表中找到某个字符串，则匹配成功，识别出一个词。匹配完成后，文档分词基本完成。

4 中文分词算法的Delphi关键代码

执行分词代码，把“成语是汉语词汇的一个重要组成部分。”这句话进行分词，分词结果为：“成语/是/汉语/词汇/的/一个/重要/组成/部分/。” 这里列出分词的二个重要算法过程。

```
procedure TCnDiv.Step1(const AStr: string; AWordList: TStrings);
//astr 为待分词的句子，awordlist 为分词结果列表
var
  ix: Integer;
  tmpWord, tmpChr, NotCnWord: string;
begin
  ix:=1;
  tmpWord:='';
  NotCnWord:='';
  while ix <= Length(AStr) do //判断是否小于待分词的句子长度
  begin
   tmpChr:=MidBStr(AStr,ix,CharLength(AStr,ix));
   //从待分词的句子取第 ix 个的字符
   if not IsStopWord(tmpChr) then //判断是否停用词
   begin
     if ByteType(tmpChr,1) = mbSingleByte then //判断是否单字节
     begin
       {先处理之前的中文部分}
       if tmpWord <> '' then
       begin
          Step2(tmpWord,AWordList);  //调用 step2 进行分词
```

```
            tmpWord:='';
        end;
        {通过检查字节类型，将非中文部分提取出来}
        while (ix <= Length(AStr)) and
(ByteType(tmpChr,1) = mbSingleByte) and (not
IsStopWord(tmpChr)) do
        begin
          NotCnWord:=NotCnWord + tmpChr;
          Inc(ix);
          tmpChr:=MidBStr(AStr,ix,CharLength
  (AStr,ix));
        end;
        {把非中文字符添加到列表}
        AWordList.Add(NotCnWord);
        NotCnWord:='';
        {清空 tmpchar}
        if tmpChr <> '' then
        begin
          Dec(ix);
          tmpChr:='';
        end;
      end
      else
        tmpWord:=tmpWord + tmpChr;  //将
  中文字符加入 tmpword
    end
    else
    begin
      {如果是停用词，则处理之前的中文部分}
      if tmpWord <> '' then
      begin
        Step2(tmpWord,AWordList);  // 调 用
step2 进行分词
        tmpWord:='';
        AWordList.Add(tmpChr); //把停用词添
加至列表
      end;
    end; //end if
    ix:=NextCharIndex(AStr,ix);  //句子下一个
字符起始字节位置
  end; //end while
  if tmpWord <> '' then
    Step2(tmpWord,AWordList);  //调用 step2
进行分词
end;
procedure TCnDiv.Step2(const AStr: string;
AWordList: TStrings);
//astr 为待分词的词条，awordlist 为分词结果列表
var
  ix: Integer;
  tmpWord, tmpChr, MaxWord: string;
begin
  ix:=1;
  tmpWord:='';
  while ix <= Length(AStr) do   //判断是否小于
待分词的词条长度
  begin
    tmpChr:=MidBStr(AStr,ix,CharLength(AStr,i
x));
    //从待分词的词条取第 ix 个的字符
    tmpWord:=tmpWord + tmpChr;
    if FDict.Find(tmpWord) > -1 then
    begin
    MaxWord:=tmpWord; //在词库中找到该词条
则赋值给 maxword
  end
  else
  begin
    if MaxWord <> '' then
    begin
      AWordList.Add(MaxWord);//  如  果
maxword 不为空，则加至列表
      MaxWord:='';
      tmpWord:=tmpChr;
    end;
    if Length(tmpWord) > FDict.MaxWordLen
then
    {处理大于设定的最大词长的词条，这里最大词
长设为 8}
    Begin
      tmpChr:=LeftBStr(tmpWord,CharLengt
h(AStr,1));
```

```
            //取超长词条的第一个字
            AWordList.Add(tmpChr);//把超长词条的第一个字加入列表
            ix:=ix - Length(tmpWord) + Length(tmpChr);
            //超长词条第二个字的字符起始字节位置，即从超长词条第二个字的位置开始匹配字库
            tmpWord:='';
          end;
      end; //end if
      ix:=NextCharIndex(AStr,ix); //词条 ix 的下一个字符起始字节位置
    end;   //end while
    if MaxWord <> '' then
      AWordList.Add(MaxWord);//把在词库找到的词条加至列表
    if (tmpWord <> '') and not AnsiSameText(tmpWord,MaxWord) then
      AWordList.Add(tmpWord); //词库找不到的词条加入列表
end;
```

5 结论

本文编写的分词词库具有最新、高频、巨大的特点，并按词条首字第一字节内码索引，采用了简单、实效的最大匹配法进行分词，提高了分词准确率，而且加快了分词速度。

参考文献

1 Steve Teixeira Xavier Pacheco .Delphi 6 开发人员指南, 北京:机械工业出版社, 2002.

2 殷建平.汉语自动分词方法.计算机工程与科学, 1998, 20(3):60－66.

3 Feng HD, Chen K, Deng XT, et al. Accessor Variety Criteria for Chinese Word Extraction.Computational Linguistics, 2004, 30(1): 75－93.