

文章编号:1001-9081(2003)06Z-0004-02

将汉字转化为拼音的研究与实现

张 炜,唐慧强

(南京气象学院 信息工程系,江苏 南京 210044)

摘 要:讨论了如何实现将全文及单个词语转化为拼音的技术,结合 Delphi 给出了具体的实现方法,并就该技术的应用进行了展望。

关键词:Delphi;拼音

中图分类号:TP311.52 **文献标识码:**A

1 引言

对于中小学校的教师以及家长和孩子在使用电脑编辑文档的时候,经常需要输入汉语拼音再加上它的声调;在日常工作和生活中我们经常使用电子记事本查找个人通信录信息,或在企业 MIS 中查询客户档案或业务资料,能够正确、快速地查询符合条件的数据,是实现各种查询功能的重点。在一般的系统开发过程中,查询功能都是通过对于字符进行比较、判断等简单方法来实现,显然无法解决上述问题。

而要实现这些功能,就必须用到拼音。目前,实现这种拼音查询的主要有两种方式:一种是利用使用 SQL SERVER 的 DTS 导入向导,将拼音码文本文件导入数据库,形成拼音字典查询临时数据库,通过字音分开最后生成拼音字典数据库;另外一种方法是手工建立对照表,每一个汉字和其拼音对应,形成数据库。

但是,这两种方法都有一个很大的弊端就是:不能将全文都转换成拼音,还有就是对单个词语的也不好控制。所以提出了一种新的方法。

我们在用拼音输入法输入时,是由拼音得到汉字,而有的输入法是支持逆向输入的,即:由汉字得到拼音。这种方法实际上就是得到汉字的字根。在本文中,采用的是微软拼音输入法(因为,该输入法是系统的默认安装设置,考虑到大多数用户已经安装,方便用户)。当然,也可以根据需求,选择合适的输入法。

2 转换文件的源程序

//在程序的开始创建时,我们判断用户的系统是否安装了微软拼音输入法,如果没有,提示用户先安装该输入法

```
procedure TForm1. FormCreate( Sender: TObject);
var
  i: integer;
begin
  II := 0;
  iHandleCount := GetKeyboardLayoutList(20, pList);
  for i := 1 to iHandleCount do
  begin
    if ImmEscape( pList[ i ], 0, IME _ESC _IME _NAME, @
      szImeName) > 0 then
      if szImeName = '微软拼音输入法' then
```

```
begin
  II := i;
  showmessage( '已安装了微软拼音');
  exit;
end;
end;
showmessage( '请安装微软拼音输入法');
end;
//首先选择需要标注拼音的文件. 如果用户不选择生成后的文件
所保存到的路径的话
//系统就自动将其保存为“待转换文件名. pytxt”文件
procedure TForm1. BtnSelectClick( Sender: TObject);
begin
  opendialog1. Title := '选择要转换的文件';
  if opendialog1. Execute then
    edit1. Text := opendialog1. FileName ;
    edit2. Text := changefileext( opendialog1. Filename, '. pytxt');
  end;
//用户也可以在这里选择生成的拼音文件所保存到的路径
procedure TForm1. BtnSaveClick( Sender: TObject);
begin
  opendialog1. Title := '转换到: ';
  if opendialog1. Execute then
    edit2. Text := opendialog1. FileName ;
  end;
//下面的函数利用输入法取得 sChinese 汉字对应的拼音
function querycompstr( hKB: HKL; const sChinese: Ansistring):
  string;
var
  dwGcl: dword;
  szbuffer: array[0..254] of char;
  iMaxkey, istart, i: integer;
begin
  result := '';
  iMaxKey := immEscape( hkb, 0, ime_esc_max_key, nil);
  if iMaxKey <= 0 then
    exit;
  //看看这个输入法是否支持 Reverse Conversion( 反向输入) 功
  能,同时,侦测需要多大的空间容纳取得的信息
  dwGcl := immgetconversionlist( hkb, 0, pchar( schinese), nil, 0,
    gcl_reverseconversion);
  if dwGcl <= 0 then
    exit;
    // 该输入法不支持 Reverse Conversion 功能
```

收稿日期:2002-10-21

作者简介:张炜(1979-),男,河南人,硕士研究生,主要研究方向:系统分析与集成,智能仪器;唐慧强(1965-),男,浙江人,副教授,主要研究方向:现代测控理论与技术集成、智能仪表等方面的研究。

```

//取得组字字根信息, dwGCL 的值必须用上次呼叫
ImmGetConversionList 得到的返回值作为参数
dwGcl := immgetconversionlist ( hkb, 0, pchar ( schinese ), @
szbuffer, dwGcl, gcl_reverseconversion );
if dwgcl > 0 then
begin
    istart := byte( szbuffer[ 24 ] );
    for i := istart to istart + imaxkey * 2 do
        appendstr( result, szbuffer[ i ] );
    end;
end;
//这段程序是实现部分, 并将结果写入文件
procedure TForm1. BtnConvertClick( Sender: TObject );
var
    f1, f2: textfile;    //f1, f2 分别存放待转换和转换后的文件
    ch1, ch2, ch11: char;
    ch2str: string;
    j, alr, tmp: integer;
    py: array[ 1..6 ] of integer;
begin
    tmp := 0;
    if not fileexists( edit1. Text ) then
        begin showmessage( '请你选定一个文件或你'#13#10'选择的文件不存在' );
            exit;
        end;
    assignfile( f1, edit1. Text );
    Reset( f1 );          //移至待转换文件的开头
    assignfile( f2, edit2. Text );
    rewrite( f2 );        //准备向欲生成的拼音文件中写入
    while not eof( f1 ) do //
        begin
            alr := 0;

```

```

read( f1, ch1 );
if not isdbcsleadbyte( byte( ch1 ) ) then
    //判断是否为汉字以外其他的字符, 若是, 直接写入
    转换后的文件中
begin
    write( f2, ch1 );
    continue;
end;
read( f1, ch11 );
ch2str := querycompstr( plist[ II ], ch1 + ch11 );
//汉字为两字节
for j := 1 to 8 do
    Write( f2, ch2str[ j ] );
    Write( f2, ' ' );
end;
closefile( f2 );
closefile( f1 );
showmessage( '转换完毕' );
end;

```

3 结束语

可以将这项技术用到中小学校的教学中以及各类会用到拼音模糊查询的管理系统中, 并且已经在电力 MIS 中得到应用。我们还可以将它扩展一下, 利用该技术并结合 HOOK 技术就可以实现类似一些翻译软件的屏幕即指即出现读音的功能(该程序在 Delphi6.0 + Windows 2000/98 中测试通过)。

参考文献

- [1] Cantu M. Delphi5 从入门到精通[M]. 王辉, 李澎东, 等译. 北京: 电子工业出版社, 2000.
- [2] Calvert C. Delphi4 编程技术内幕[M]. 北京: 机械工业出版社, 1999.

(上接第3页)

```

{
    switch( token )
    {
        selectsymbols( < if_stmt > ): if_stmt();
        break;
        selectsymbols( < repeat_stmt > ): repeat_stmt();
        break; selectsymbols( < assign_stmt > ): assign_stmt();
        break;
        selectsymbols( < read_stmt > ): read_stmt();
        break;
        selectsymbols( < write_stmt > ): write_stmt();
        break;
    }
}
void if_stmt()
{
    match( "if" );
    exp();
    match( "then" );
    stmt_sequence();
    if ( token == selectsymbols( 'else' < stmt_sequence > ) )
    {
        match( "else" );
        stmt_sequence();
    }
}
.....

```

这里 selectsymbols(string α) 是一个计算串 α 的 Select 集合的函数, 其返回值是一个字符串数组, 保存了 Select 集中的各个单词, 可以从 α 的 First 集和 Follow 集计算等到, 算法已经给出, 这里就不再给出实现细节了, 并且为了简便起见这里并没给出形如 “token == selectsymbols(' ; ' < statement >)” 该类语句的最终形式: “token == SEMI”, 这里 SEMI 是分号所对应的整数类型的内部码。

5 结语

著名的 YACC(Yet Another Compiler Compiler) 是采用自底向上的句法分析方法, 自动生成 LALR(1) 文法的句法分析程序。本文给出了采用自顶向下的句法分析方法自动生成 LL(1) 文法句法分析程序的算法及其实现, 文中没有涉及语义部分的讨论, 事实上一旦识别出一个正确的句子, 则可以插入相应的语义。该自动生成程序可用于教学目的使用, 亦可以给需要用 LL(1) 文法进行句法分析的程序编写者带来帮助。

参考文献

- [1] (美) Louden KC. 编译原理及实践[M]. 冯博琴, 等译. 北京: 机械工业出版社, 2000. 1-148.
- [2] 徐国定, 杨宗源. 编译程序构造[M]. 上海: 华东师范大学出版社, 1989. 84-118.
- [3] Richter J. Applied . Net Framework Programming[M], 2002.