ICTCLAS 中文词法分析的 Delphi 调用研究

郭晓云
(南京政治学院上海分院信息管理系，上海 200433)

摘 要：深入研究了 ICTCLAS2011 的 API，并利用 Delphi 实现了对其所有 API 的调用。

关键词：ICTCLAS；中文分词；Delphi

Research and Implementation of ICTCLAS API with Delphi

GUO Xiaoyun
(Shanghai Branch Nanjing Institute of Politics , Shanghai 200433)

Abstract: This paper researches the ICTCLAS2011 API and implement all API functions in Delphi.

Key words: ICTCLAS ; Chinese Word Segmentation ; Delphi

ICTCLAS 中科院计算所汉语词法分析系统是基于层次隐马尔科夫模型的中文词法分析系统。系统提供中文分词、词性标注、命名实体识别、新词识别等功能，支持用户词典，支持多种编码格式，是目前业界公认的最好的汉语词法分析器，广泛应用于中文信息处理的各个领域[1]。

ICTCLAS 提供了 C++、C#、Java 3 种语言的接口文件和调用文档，虽然官方文档宣称支持 Delphi，但并没有提供 Delphi 的接口文件和文档，导致 Delphi 程序员调用 ICTCLAS 时缺乏必要的头文件和调用文档参考。为了解决这个问题，本文对最新版的 ICTCLAS2011（build 0404）的文档和程序进行了细致的研究，利用 Delphi 对 ICTCLAS 接口进行了封装，并给出了每个 API 函数的详细说明。

1 Delphi 调用 DLL 的方法

ICTCLAS 采用 C++编码，在 Windows 中以 DLL 动态链接库形式提供调用。对 DLL，Delphi 提供了两种调用方式：静态调用和动态调用[2]。

静态调用将 DLL 的导出函数当做 Delphi 的普通函数处理，使用 External 关键字直接声明和导入 DLL 函数。DLL 的加载和释放由系统处理。静态调用非常方便，但程序高度依赖 DLL 文件，如果 DLL 出错，比如 DLL 文件路径不对、DLL 声明错误，程序将无法启动。

动态调用在程序运行时使用 Windows API 函数 LoadLibrary 动态加载 DLL，使用 GetProcAddress 获取 DLL 函数的指针地址，再通过指针调用函数。当不需要使用 DLL 时，使用 FreeLibrary 动态释放 DLL。动态调用灵活可控，可按程序需要动态加载动态释放，即使 DLL 出错也不会导致系统无法运行，但比静态调用要复杂。

对 ICTCLAS，静态和动态两种调用方式都适用。篇幅所限，主要介绍静态调用方法，动态调用仅给出一个小例子。程序在 Delphi2006 和 Delphi2010 中都编译通过。

2 ICTCLAS 的组成及使用方法

ICTCLAS2011 由主文件 ICTCLAS2011.dll、配置文件 configure.xml、许可证文件 license.dll 及数据文件夹 Data 组成。数据文件夹中存放模型文件、词典文件、词表文件、词性标注集映射文件等。

ICTCLAS2011 的 API 由十几个 API 调用函数组成，大致可分为初始化与卸载、文本文件处理、文本段落处理、字典处理及辅助功能 5 类函数。应用程序首先需要调用 ICTCLAS_Init 函数进行模型初始化，包括读取配置文件、加载字典、建立内存对象等操作；然后再调用功能函数进行分词、词性标注、字典管理、关键词抽取等处理；最后调用 ICTCLAS_Exit 函数退出模型，释放占用的资源。

3 ICTCLAS 的 Delphi 调用方法

Delphi 静态调用 DLL 时，External 声明的函数形式必须与 DLL 的导出函数形式完全一致，否则在程序运行时会报 XX 函数找不到的错误。由于 ICTCLAS2011 在编译时并没有指定 Def 文件，C++在编译时进行了多态处理，导致 ICTCLAS2011 的 DLL 导出函数并不是其附带的接口文档中的形式，而是形如“?ICTCLAS_Init@@YA_NPBDH@Z”这样的形式。很多 Delphi 程序员正是由于这个原因，而无法正常使用 ICTCLAS2011。

解决这个问题需要使用可查看 DLL 导出函数的 PE 工具，推荐使用 Visual Studio 提供的 Dependency Walker[3]。Depends 会显示出 DLL 导出函数的列表，包括函数的序号、函数名。在声明函数时，可直接采用序号的方式声明函数，也可用工具显示的函数名声明。

3.1 ICTCLAS 常量和数据结构定义

ICTCLAS 定义了一些常量和数据结构。常量包括标注集代码、编码代码等，常量定义可参考 ICTCLAS 提供的 C++头文件。除了常量，还有一个很重要的数据结构 Result_t，用于存放分词及词性标注的结果，每个对象存放对一个词的处理结果，此结构定义如下：

```
result_t = record
  start: integer; //词语在输入句子中的开始位置
  length: integer; //词语的长度
  iPOS: integer; //词性
```

作者简介：郭晓云（1977-），男，讲师，研究方向：信息检索、管理信息系统。

收稿日期：2011-10-19

```
    sPos: array [0..POS_SIZE - 1] of ansichar;// 词性编号
    word_ID: integer;
    //词在字典中的编号，如果是未登录词，则为 0
    word_type: integer; //词是否在用户词典中;1,是用户词典
//中的词;0,非用户词典中的词
    weight: integer; //词的权重，抽取关键词用
  end;
  Presult_t = ^result_t;//定义一个结构指针
```

3.2 ICTCLAS 的初始化与卸载

初始化 ICTCLAS 使用 ICTCLAS_Init 函数，卸载使用 ICTCLAS_Exit 函数。

两个函数的静态调用方法如下：

首先在单元的声明部分声明如下函数，这里假定程序和 DLL 文件在一个文件夹中。

```
function ICTCLAS_Init (const pszInitDir: PAnsiChar = nil;
encode: Integer = GBK_CODE) : BOOL; cdecl;// pszInitDir 指定
//初始化的文件夹，encode 指定文本编码类型
function ICTCLAS_Exit () : BOOL; cdecl;
```

注意，由于 ICTCLAS 的头文件中函数声明为 cdecl 方式[4]，所以参数调用方式应选择 cdecl。C++中的字符串变量在 Delphi 中应定义为 Pansichar，而不是 Pchar。因为在 Delphi2010 中，为了处理 Unicode，默认的 Pchar 已修改为 4 字节的 Pwidechar。

然后在单元的 Implementation 部分增加函数与 DLL 的映射。

```
function ICTCLAS_Init; external 'ICTCLAS2011.dll' name '?
ICTCLAS_Init@@YA_NPBDH@Z';
function ICTCLAS_Exit; external 'ICTCLAS2011.dll' name '?
ICTCLAS_Exit@@YA_NXZ';
```

如果采用序号方式，以 ICTCLAS_Init 为例，其实现方法如下：

```
function ICTCLAS_Init; external 'ICTCLAS2011.dll' index 15;
```

两个函数的返回值都为 BOOL 类型，成功返回 true，失败返回 false。

如果采用动态调用方法，以 ICTCLAS_Init 函数为例，其过程如下：

```
Tinit = function (const pszInitDir: PAnsiChar = nil; encode:
Integer = GBK_CODE) : BOOL; cdecl; //声明函数类型
Var Func:Tinit; //首先定义一个函数变量
Fhandle := loadlibrary ('ICTCLAS2011.dll') ;
//然后动态加载 DLL 文件
if Fhandle > HINSTANCE_ERROR then //如果加载成功
begin
  @func := GetProcAddress (Fhandle, lpcstr (15)) ;
  //以序号方式获取函数指针
  //@func := GetProcAddress (Fhandle,'?
ICTCLAS_Init@@YA_NPBDH@Z') ; //以函数名形式
if Assigned (@func) then Func () ;
//如果获得了函数指针，则通过指针调用此导出函数。
  FreeLibrary (Fhandle) ;//最后释放 DLL 文件。
end;
```

3.3 ICTCLAS 处理文本文件

ICTCLAS 处理文本文件使用 ICTCLAS_FileProcess 和 ICTCLAS_FileProcessEx 两个函数。前者默认进行词性标注，后者则可由程序指定。

两个函数的声明如下：

```
function ICTCLAS_FileProcess ( const sSrcFilename:
PAnsiChar; const sDsnFilename: PAnsiChar; const bPOStagged:
integer = 1) : double; cdecl;
function ICTCLAS_FileProcessEx (const sSourceFilename:
PAnsiChar; const sResultFilename: PAnsiChar) : double; cdecl;
```

处理文本文件时直接调用此函数即可，其中参数 sSrcFilename 指定要处理的文本文件名，sDsnFilename 指定结果文件名，bPOStagged 指定是否词性标注，取 1 标注，0 则不标注。函数返回一个双精度实数错误码，如果成功则为 1000000。

3.4 ICTCLAS 处理文本段落

ICTCLAS 处理文本段落比直接处理文本文件复杂，需要 API 函数与 Result_t 数据结构配合，函数包括 ICTCLAS_ParagraphProcess、ICTCLAS_ParagraphProcessA。其中 ICTCLAS_ParagraphProcess 直接返回字符串形式的分词结果，ICTCLAS_ParagraphProcessA 则返回一组代表词的 Result_t 数据结构。

两个函数的声明如下：

```
function ICTCLAS_ParagraphProcess ( const sParagraph:
PAnsiChar; const bPOStagged: integer = 1) : PAnsiChar; cdecl;
function ICTCLAS_ParagraphProcessA ( const sParagraph:
PAnsiChar; var pResultCount: integer) : Presult_t; cdecl;
//返回 Result_t 指针，指针指向数量为 pResultCount 的 Result_t
//变量，变量内存由 ICTCLAS 处理，不需要程序员去释放。
```

下面是调用 ICTCLAS_ParagraphProcessA 的例子：

```
count := 0;//先定义一个整数变量 count
Pre:= ICTCLAS_ParagraphProcessA (pansichar
(paragraph) , count) ; // 对 paragraph 字符串变量进行处理，
//Pre 是一个 Presult_t 变量。函数运行后，将修改 count 变量为
//产生的 result_t 结构数量。
for I := 0 to count - 1 do//对 Pre 进行处理，结果存放在一
//个 Tmemo 控件中
  begin
    memoresult.Lines.Add (inttostr (Pre.start)) ;
    //词语在输入句子中的开始位置
    memoresult.Lines.Add (inttostr (Pre.length)) ;
    //词语的长度，注意每个汉字的长度为 2
    memoresult.Lines.Add (inttostr (Pre.iPOS)) ; //词的词性
    memoresult.Lines.Add (inttostr (Pre.word_ID)) ;
    //词在词典中的编号，0 代表未登录词
    memoresult.Lines.Add (inttostr (Pre.word_type)) ;
    //1 代表用户词典中的词；0 代表非用户词典中的词
    memoresult.Lines.Add ('————') ;//分隔符
    Inc (Pre) ;//指针递进，处理下一个词
  end;
```

3.5 ICTCLAS 的用户词典处理

词典的质量直接决定最后的分词结果，ICTCLAS 的词典

（下转到 18 页）

或者将其关闭。毕竟集团公司是以利润最大化为目的，只有利润提升才能提高各个单位的生产积极性，才能有财力去支持新技术的研发和市场开拓。

表 1 子企业的初始分配结果集

i \ j	1	2	3	4	5
1	BAC	CA	B	B	C
2	BAC	AC	B	B	C
3	BAC	CA	B	B	C
4	BAC	CA	B		C
5	BAC	AC	B	B	C

表 2 最终的企业订单分配结果集

i \ j	1	2	3	4	5
1	B	A	B	B	
2	C	A	B	B	C
3	C	A	B	B	C
4	A	C	B		C
5	B	A	B	B	C

5 结语

对于订单分配模型的应用不能仅仅是用它来进行安排生产，应该进一步挖掘它的用途。它可以为企业集团的重新组合通过实际的生产安排得出数据，为集团进行重组提供强有力的数据依据。

参考文献

[1] 郑海航，等. 中国企业兼并研究 [M]. 北京：经济管理出版社，1999.

[2] 才婉如. 90 年代欧美企业兼并浪潮及其对中国的启示 [J]. 宏观经济研究，2000，(5).

[3] 程兆谦. 购并整合七法则 [J]. 中外管理，2001，(3).

[4] 李时椿. 中外成功企业并购重组的策略研究 [J]. 经济管理，2001，(10).

[5] 刘晓冰，王宇春. 钢铁企业集团订单分配模型研究 [J]. 西南交通大学学报，2006，(2).

[6] Xudongsong，Djun zhu.Study on enterprise group order allocation multiobjective model,JCIT，2010 (5).

(上接第 11 页)

处理方式可参看《ICTCLAS 代码学习笔记》[5] 一文。ICTCLAS 支持用户自定义词典，用户可以在词典中加入自己的词条。词典处理包括 4 个函数，其定义如下：

```
function ICTCLAS_ImportUserDict ( const pszFileName: PAnsiChar) : integer; cdecl;//导入文本格式的词典，每行一个
//词条。pszFileName 指定文本文件名，返回导入的词条数量。
function ICTCLAS_AddUserWord ( const sWord: PAnsiChar) : integer; cdecl;//在内存词典中动态增加一个词
//sWord，成功返回 1，失败返回 0。
function ICTCLAS_DelUsrWord ( const sWord: PAnsiChar) : integer; cdecl;//在内存词典中动态删除 sWord 词
//条。如果词条不存在，返回-1，否则返回该词条的编号。
function ICTCLAS_SaveTheUsrDic () : integer; cdecl;
//将内存中的用户词典保存在 data 目录下的 UserDict.pdat 中，
//成功返回 1，失败返回 0。
```

3.6 其他辅助功能

ICTCLAS 还提供了一些附加功能，包括取词的 Unigram 概率、段落关键词、段落文本指纹等。其声明如下：

```
function ICTCLAS_GetUniProb ( const sWord: PAnsiChar) : double; cdecl;
//返回 sWord 的 Unigram 概率（双精度实数值）
function ICTCLAS_IsWord ( const sWord: PAnsiChar) : integer; cdecl;//判断词是否在字典中
function ICTCLAS_KeyWord ( resultKey: Presult_t; var nCountKey: integer) : integer; cdecl;//取段落关键词，结果存放
//在 result_t 数组中，数组的维度在 nCountkey 中，词按其权重
//降序排列。函数成功返回 1，失败返回 0。
function ICTCLAS_FingerPrint () : longint; cdecl; //文本指
//纹提取，须在 ICTCLAS_ParagraphProcessA 函数执行完后执
//行，返回一个整数。
function ICTCLAS_SetPOSmap (const nPOSmap: integer) : integer; cdecl; //指定词性标注集
```

取关键词和文本指纹的例子程序：

```
count := 0;
ICTCLAS_ParagraphProcessA ( pansichar ( paragraph) , count) ; //首先得到段落分词后的词数
code := ICTCLAS_FingerPrint () ;//取文本指纹
memoresult.Lines.Add (IntToHex (code, 8)) ;
//按 16 进制显示
getmem (resultKey, sizeof (result_t) * count) ;
//创建 result_t 数组
code := ICTCLAS_KeyWord (resultKey, count) ;
//抽取关键词，1 成功 0 失败
if code > 0 then
 for I := 0 to count - 1 do
  begin
   word := copy (paragraph, resultKey.start + 1, resultKey.length) ;//在段落中找到这个词
   memoresult.Lines.Add (word + ' ' + inttostr (resultKey.weight)) ;//显示词的权重
   Inc (resultKey) ;//定位到下一个词对象
  end;
 freemem (resultKey) ; //释放对象数组
```

参考文献

[1] 张华平. ICTCLAS2011 接口文档. 北京理工大学计算机语言信息处理研究所，2011.

[2] 刘艺. Delphi 面向对象编程思想. 北京：机械工业出版社，2004.

[3] Dependency Walker.http://www.dependencywalker.com/.2011.

[4] 张华平. ICTCLAS2011 Windows 下 C 接口. 北京理工大学计算机语言信息处理研究所，2011.

[5] 黄瑾. ICTCLAS 代码学习笔记. 中科院计算技术研究所多语言交互技术评测实验室，2006.

ICTCLAS中文词法分析的Delphi调用研究

作者： 郭晓云， GUO Xiaoyun
作者单位： 南京政治学院上海分院信息管理系,上海,200433
刊名： 电脑编程技巧与维护
英文刊名： Computer Programming Skills & Maintenance
年，卷(期)： 2011(24)
被引用次数： 2次

参考文献(5条)

1. 张华平 ICTCLAS2011接口文档 2011

2. 刘艺 Delphi面向对象编程思想 2004

3. Dependency Walker 2011

4. 张华平 ICTCLAS201 1 Windows下C接口 2011

5. 黄瑾 ICTCLAS代码学习笔记 2006