文章编号:1001 - 9081(2004)12Z - 0177 - 02

利用 JNI 实现 ICTCLAS 系统的 Java 调用

夏 天,樊孝忠,刘 林
(北京理工大学 计算机科学与技术系,北京 100081)
(xiatian@bit.edu.cn)

摘 要:介绍了 Java 本地方法的作用及意义,详细讨论了在 Windows 平台上,利用 JNI(Java Native Interface)在 Java 中调用 ICTCLAS 系统进行词法分析的具体方法和整个过程,并阐述了 JNI 实施过程当中的一些注意事项。

关键词:Java 本地接口;本地方法;ICTCLAS;词法分析
中图分类号: TP311.52 **文献标识码**:A

0 引言

词是最小的、能够独立活动的、有意义的语言成分[1],汉语则以字为基本的书写单位,词和词之间没有明显的分割标志,因此,进行中文词法分析是中文信息处理的基础与关键,是开展相关研究所必需的一个重要环节。如今,对汉语分词和词性标注的研究已经达到了一个比较成熟的阶段,很多分词程序的正确率已经可以达到 95% 以上,国内许多科研单位也研制出了各具特色的实用系统[2]。其中,由中国科学院计算技术研究所的张华平先生设计开发的分词和词性标注一体化系统(Institute of Computing Technology, Chinese Lexical Analysis System, ICTCLAS),分词正确率高达 97.58%,未登录词识别召回率均高于 90%,其中中国人名的识别召回率接近 98%,处理速度为 31.5Kbytes/s,实现了分词与词性标注的一体化,未登录词与普通词处理的一体化,评估体系一体化,具有较高的实践性与正确性。

ICTCLAS 采取 HMM 模型,建立切分词图。在词语粗分阶段,先得出 N 个概率最大的切分结果,然后,利用角色标注方法识别未登录词,并计算其概率,将未登录词加入到切分词图中,之后视它为普通词处理,最终进行动态规划优选出 N 个最大概率切分标注结果[3]。系统由 C++ 语言实现,速度快,准确率高,并提供了一套完整的动态连接库 ICTCLAS.dll 和相应的概率词典,开发者可以完全忽略汉语词法分析,直接在自己的系统中调用 ICTCLAS,在分词和词性标注的基础上继续上层开发。

近年来,随着网络的快速发展,越来越多的研究人员开始采用面向对象的 Java 语言进行编程,由于 Java 语言具有简单、面向对象、面向网络、平台无关、安全性、多线程、动态性等诸多优点[4],我们在开发面向金融领域的自动问答系统[5]时,就以 Java 作为主要编程语言,采用 JNI 技术,实现了问答核心模块与 ICTCLAS 词法分析模块的无缝链接。

1 JNI 简介

JNI 是 Java 与其他编程语言的集成编程接口,又称为本地方法接口,它允许运行在虚拟机上的 Java 程序调用其他语言(例如 C 或 C++)编写的程序或类库,也能够将 Java 虚拟机直接嵌入到本地的应用程序当中,允许本地方法创建、检查及更新 Java 对象,调用 Java 方法,引用 Java 类,捕捉和抛出异常等。JNI 的基本结构描述如图 1 所述。

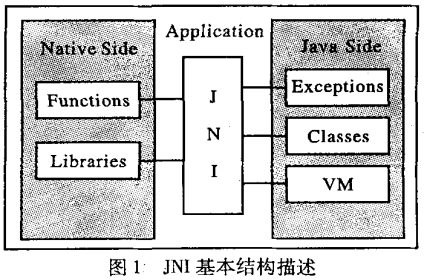

图 1 JNI 基本结构描述

JNI 具有二进制兼容、效率高和功能强三个显著特点[6],它的引入,使 Java 与其他语言的互操作成为可能。当然,在软件系统中,使用 JNI 方法通常会受到一定程度的限制,例如无法在 Java 小应用程序中使用 JNI,使用 JNI 后,由于其他语言(如 C/C++)可能能够随意地分配对象、占用内存,Java 的指针安全性得不到保证,代码可移植性也受到一定程度的挑战。但在有些情况下,使用 JNI 是可以接受的,甚至是必须的,比如,使用一些旧的库,与硬件、操作系统进行交互,或者为了提高程序的运行速度等。

我们之所以采用 JNI 调用 ICTCLAS 的技术方案,是基于以下两方面的考虑:

1) 词法分析模块作为中文信息处理系统中的一个基础模块,运行速度至关重要,用 C/C++ 编写的程序运行速度要比用纯 Java 编写的程序效率更高,速度更快;

2) 如果重写源代码,把 ICTCLAS 系统移植到 Java 环境中,开发和测试的工作量都很大,耗时费力,利用本地方法及其所提供的接口,把现有程序移植到 Java 平台将会更容易,也更利于分工合作。

2 实现过程

系统的整体结构如图 2 所示。可以看到,在调用 ICTCLAS 系统时,并不能直接使用现成的动态库文件 ictclas.dll,而是采用了一个遵循 JNI 规范的中间层,由中间层实现对 ICTCLAS 接口函数的封装,同时负责与相关的 Java 类交互。下面以 Windows2000 为系统实现平台,以在 Java 中调用 ICTCLAS 词法分析系统为例,详细说明 JNI 的具体实现过程。

收稿日期:2004 - 03 - 04

作者简介:夏天(1978 -),男,山东潍坊人,博士研究生,主要研究方向:自然语言处理; 樊孝忠(1948 -),男,河南人,教授,博士生导师,主要研究方向:自然语言处理、多媒体网络教学; 刘林(1974 -),男,辽宁人,博士研究生,主要研究方向:信息提取.

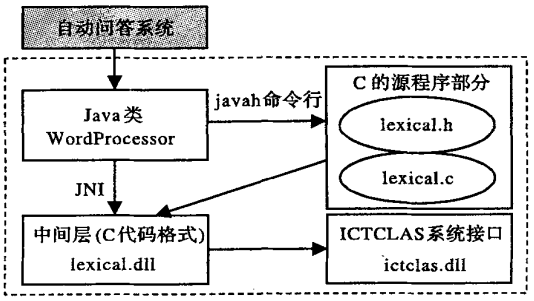

图2 系统结构

1) 访问本地方法的 Java 代码

首先定义一个 WordProcessor 类,在该类中对需要用到的本地方法进行声明,本地方法的声明必须使用关键字"native",而且不能拥有方法主体,实现部分放在了 Java 类之外的本地代码之中,在 Windows 平台下将编译成为一个 DLL 文件(Unix/Linux 下为一个 so 文件)。

```
package bitnlp. lexical;
public class WordProcessor {
  public static String segment( String sentence) {
    String result = "";
    try{ result = new String( getWordSegment
        ( sentence. getBytes( "gb2312" ))) ;
    } catch( Exception e) { result = e. getMessage( ) ; }
    return result; }
  public static native boolean initialLib( ) ;
  public static native byte[ ] getWordSegment( byte[ ] sentence) ;
  ……                                          //其他本地方法声明
  static{
    System. loadLibrary( "lexical") ;
    WordProcessor. initialLib( ) ; } }
```

当 Java 程序开始运行时,通过 System. loadLibrary( ) 方法加载包含本地代码的共享库,上例中还紧跟着调用本地方法对词法分析系统进行了必要的初始化处理。需要说明的是,在 C/C++ 中,常在函数中传递指针参数,但由于 Java 中没有指针,因此可以在 Java 中采用传引用的方式来处理类似 C/C++ 中指针的问题。另外,为了解决中文乱码问题,我们采用 byte[ ]类型来取代常用的 String 类型作为本地方法的参数声明,同时增加一个新方法实现由 String 到 byte[ ] 的转化和对本地方法的调用封装。

2) 编译 Java 类

执行命令 javac bitnlp \ lexical \ WordProcessor. java,编译 WordProcessor 类生成 WordProcessor. class 文件。

3) 创建 C/C++ 头文件

执行命令 javah-jni-o lexical. h bitnlp. lexical. WordProcessor,产生包含本地方法原型的头文件 lexical. h,去掉程序注释后的文件代码如下:

```
#include <jni. h >
#ifndef _Included_bitnlp_lexical_WordProcessor
#define _Included_bitnlp_lexical_WordProcessor
#ifdef __cplusplus
extern "C" {
#endif
JNIEXPORT jbyteArray JNICALL
 Java_bitnlp_lexical_WordProcessor_getWordSegment
 (JNIEnv *, jclass, jbyteArray) ;
JNIEXPORT jboolean JNICALL
 Java_bitnlp_lexical_WordProcessor_initialLib (JNIEnv *, jclass) ;
 ……                                          //其他本地方法函数声明
#ifdef __cplusplus}
#endif
#endif
```

其中,JNIEXPORT 和 JNICALL 是用于导出函数的、依赖于编译器的指示符。方法中的第一个参数是 JNI 接口指针,它是一个指向函数指针表的指针,因此必须在每个 JNI 函数访问前面加前缀(* env) - >,以确保对函数指针的间接引用。函数中的第二个参数取决于此方法是否为静态方法,静态方法的第二个参数是调用本地方法的 Java 类,而非静态本地方法的第二个参数则是调用本地方法的 Java 类所属对象,剩余参数与 Java 方法中的参数相对应。

该头文件还给出了 JNI 对本地方法的所有函数说明,函数说明的形式由 JNI 的命名规范所决定[7],其命名规范遵循以下规则:

a) 使用 Java 方法名的全称,如果这个类在一个包中,还要在方法前加上包的名称,如 bitnlp. lexical. WordProcessor. getWordSegment;

b) 用下划线替代句号,并加上前缀 Java_;

c) 如果类名中包含非 ASCII 字符或数字字符,像'_','$'或带有大于'\u007F'的 Unicode 字符,用_0xxxx 来代替它们,xxxx 是四个十六进制位,它代表相应字符的 Unicode 值。

4) 实现本地方法,完成对 ICTCLAS 系统调用的具体封装

编写 C/C++ 函数实现时,函数声明要和 lexical. h 中的声明完全一致。以下是 lexical. c 文件的部分内容和相应注释:

```
#include "windows. h"
#include "lexical. h"
#define CURRENT_ENCODE "GBK"
typedef LPSTR(_cdecl * ICTCLAS_SEGMENT) (LPSTR input) ;
HINSTANCE hDLL = NULL;
ICTCLAS_SEGMENT ICTCLAS_Segment = NULL;
int bLoadFlag = 0;
JNIEXPORT jboolean JNICALL
  Java_bitnlp_lexical_WordProcessor_initialLib
(JNIEnv * env, jclass obj) {
  hDLL = LoadLibrary( "ictclas. dll") ;
                                  //加载 ICTCLAS 词法分析库文件
  if( hDLL)
  { ……                                        //初始化处理
    //以下记录了 ICTCLAS 中的分词方法句柄,为便于说明
    //内部方法名称与官方文档有所不同
    ICTCLAS_Segment = (ICTCLAS_SEGMENT)
      GetProcAddress( hDLL, "ParagraphProcess") ; }
  if( ICTCLAS_Segment! = NULL) return 1; else return 0; }
……                                            //部分辅助代码
JNIEXPORT jbyteArray
 JNICALL Java_bitnlp_lexical_WordProcessor_getWordSegment
 (JNIEnv * env, jclass obj, jbyteArray array) {
 char * result;
 char * rtn = jbyteArrayToChars( env, array) ;
 //jbtyeArrayToChar 为一自定义函数
 //实现 jbyteArray 到字符串类型的转换
 if ( ICTCLAS_Segment! = NULL)
   result = ICTCLAS_Segment( rtn) ;
 else
```

(下转第 182 页)

表 3 两个不同兴趣的用户输入"计算机"后 PISE 返回的结果分析

次序	用户 A		用户 B	
	初始 url 集	T/N	初始 url 集	T/N
1	http://www.edu.cn http://www.ccw.com.cn	10/5	http://www.onlinedown.net http://www.download.com.cn	10/7
2	http://www.edu.cn http://www.ccw.com.cn http://www.computerworld.com.cn http://www.ciw.com.cn	5/11	http://www.onlinedown.net http://www.download.com.cn http://www.pconline.com.cn/download/ http://www.soft999.com	5/13
3	http://www.edu.cn http://www.ccw.com.cn http://www.computerworld.com.cn http://www.ciw.com.cn http://www.gd.edu.cn http://www.nrcce.com	2/18	http://www.onlinedown.net http://www.onlinedown.net http://www.pconline.com.cn/download/ http://www.soft999.com http://dl.163.com/ http://download.lycos.com.cn/	2/26

注:T 为爬行时间(分钟),N 为爬行结果(网页数)。

参考文献:

[1] (美)HEATON J. 网络机器人 Java 编程指南[M]. 童兆丰,李纯,刘润杰,译. 北京:电子工业出版社,2002.

[2] 推广中国网. 搜索引擎发展史[DB/OL]. http://www.asp169.com/nous1.htm,2004-01-25.

[3] 陈奇. 面向对象程序设计高级教程[M]. 北京:高等教育出版社,2001.

[4] 吴华香,钟少丹. 链接分析法-网络计量学方法初探[J]. 情报科学,2002,20(1).

[5] Member of the Clever Project. HyperSearching the Web[J]. Science American,1998,280(6).

(上接第 178 页)

```
    result = rtn;
    return CharsTojbyteArray(env, result); }
    //CharsTojbyteArray 为一自定义函数
    //实现字符串到 jbyteArray 类型的转换
```

5) 编译生成动态链接库 lexical.dll

本地方法编写完后,还需要编译生成动态链接库才能在 Java 中调用,在 Win32 平台中,使用 Microsoft Visual C++ 编译器执行如下指令:

```
cl -I c:\jdk1.4\include -I c:\jdk1.4\include\win32 -LD
  lexical.c -Fe lexical.dll
```

其中,c:\jdk1.4 是 JDK 在本机的安装路径,在 include 和 include\win32 目录下面有产生动态连接库所需要包括的头文件。由该指令生成 lexical.dll 文件后,还需要把它所在的目录包含在系统路径中,之后就可以在 Java 中通过 WordProcessor 类中的方法,调用 ICTCLAS 系统中的函数对汉语句子进行词法分析了。

6) 运行 Java 程序测试结果

为便于测试,在 WordProcessor 类中创建一个 main 方法来调用 segment 方法,如下:

```
public static void main(String[] argv) {
  System.out.println(WordProcessor.segment
    ("这是一个测试句子."));}
```

在运行程序之前,必须确认把 lexical.dll 和 ictclas.dll 两个动态链接库文件所在的目录包含在系统路径中,而且 ICTCLAS 也能找到其概率词典的位置,然后重新编译运行 WordProcessor 类,最后输出结果:这/r 是/v 一个/m 测试/vn 句子/n 。/w。

3 实验结果与分析

利用上述方法在 Java 语言中成功地调用了 ICTCLAS 系统中的功能模块,实现了对汉语句子的自动分词和词性标注,但在连续对 50 个以上的句子进行处理时,经常会出现访问冲突异常,经分析是由于分词速度不匹配造成的,引入共享锁机制之后,该问题得到解决。另外,利用该接口对大规模领域文本进行词法分析时,Java 虚拟机在运行一段时间后(一般为 4~5个小时),有时会抛出内存访问异常,笔者认为原因在于 ICTCLAS 系统在内存管理方面存在问题,接口程序采取了在捕捉到异常后重新加载 ICTCLAS 系统的策略,以弥补该缺陷。经过改进之后的接口程序具有了故障捕捉和自动恢复功能,在对 175M 领域文本进行连续处理时,运行比较稳定,没有出现虚拟机崩溃现象,完全可以满足研究和实验之用。

4 结语

汉语词法分析系统是中文信息处理的基石,利用 JNI 实现对 ICTCLAS 系统的 Java 调用,可以帮助中文信息处理研究人员更加专注于自己的研究,减少不必要的重复劳动。实验结果也证明,JNI 是把 Java 程序与非 Java 语言程序和平台相关功能结合的有效方式,在较少考虑安全性和可移植性的情况下,JNI 对于大量已有非 Java 软件的利用,以及直接操纵硬件方面的程序编写等都将发挥很好的作用。

参考文献:

[1] 朱德熙. 语法讲义[M]. 北京:商务印书馆,1982.

[2] 王科,高常波,翟雪峰. 汉语分词的主要技术及其应用展望[J]. 通信技术,2003,(6):12-15.

[3] 张华平,刘群. 基于 N-最短路径方法的中文词语粗分模型[J]. 中文信息学报,2002,(5):1-7.

[4] HORSTMANN CS, CORNELL G. Java2 核心技术 卷 2:高级特性[M]. 北京:机械工业出版社,2000.

[5] LI HQ, FAN XZ, *et al*. The Study and Implementation of Finance-domain Chinese Automatic Question-Answering System: FAQAS[A]. Advances in Computation of Oriental Languages[C], 2003. 483-489.

[6] 李亚东,夏雨佳,席裕庚. 基于 JNI 的跨平台软件设计[J]. 计算机工程,2000,(9):87-88.

[7] STEARNS B. Trail: Java Native Interface[EB/OL]. http://java.sun.com/docs/books/tutorial/native1.1/index.html, 2000.

利用JNI实现ICTCLAS系统的Java调用

作者： 夏天， 樊孝忠， 刘林
作者单位： 北京理工大学,计算机科学与技术系,北京,100081
刊名： 计算机应用 ISTIC PKU
英文刊名： JOURNAL OF COMPUTER APPLICATIONS
年，卷(期)： 2004,24(z2)
被引用次数： 12次

参考文献(5条)

1. Horstmann CS;CORNELL G Java2核心技术卷2:高级特性 2000

2. 张华平;刘群 基于N--最短路径方法的中文词语粗分模型[期刊论文]-中文信息学报 2002(05)

3. 李亚东;夏雨佳;席裕庚 基于JNI的跨平台软件设计[期刊论文]-计算机工程 2000(09)

4. 王科;高常波;翟雪峰 汉语分词的主要技术及其应用展望[期刊论文]-通信技术 2003(06)

5. 朱德熙 语法讲义 1982

本文读者也读过(2条)

1. 蔡小艳.寇应展.沈巍.郑伟.CAI Xiao-yan.KOU Ying-zhan.SHEN Wei.ZHEN Wei 汉语词法分析系统ICTCLAS在Nutch-0.9中的应用与实现[期刊论文]-军械工程学院学报2008,20(5)

2. 刘克强 2009共享版ICTCLAS的分析与使用[期刊论文]-科教文汇2009(22)

引证文献(12条)

1. 申莹.徐东平.庞俊 基于概念的中文博客情感极性聚类分析[期刊论文]-计算机系统应用 2011(8)

2. 陶秋香.喻金科.涂继亮 基于向量空间模型的公文分类系统研究与实现[期刊论文]-南昌航空大学学报（自然科学版） 2009(4)

3. 向阳.张波.韩婕 Agent驱动的中文本体智能构建研究[期刊论文]-计算机工程与应用 2009(10)

4. 姜彩红.乔晓东.朱礼军 基于本体的专利摘要知识抽取[期刊论文]-现代图书情报技术 2009(2)

5. 周锦程.王丹 基于Lucene的全文搜索引擎研究与应用[期刊论文]-黔南民族师范学院学报 2009(3)

6. 刘洁.刘建勋 基于用户兴趣模型的Web服务发现系统设计[期刊论文]-湘潭大学自然科学学报 2008(1)

7. 姜彩红.乔晓东.朱礼军.桂婕.张运良 基于GATE的中文专利摘要的抽取[期刊论文]-数字图书馆论坛 2008(11)

8. 谢枫平 一个基于朴素贝叶斯方法的RSS分类器[期刊论文]-闽西职业技术学院学报 2008(4)

9. 齐燕.陈海 基于本体和Lucene的电子公文查询系统的研究与实现[期刊论文]-计算机与现代化 2007(2)

10. 易爱平.廖祝华.张惠 基于Google的个性化搜索系统的设计与实现[期刊论文]-电脑知识与技术（学术交流） 2007(1)

11. 曹强 基于Lucene的Web站点站内全文检索系统的设计与实现[期刊论文]-图书情报工作 2007(9)

12. 廖祝华.刘建勋.易爱平 基于用户兴趣的Web服务发现[期刊论文]-微电子学与计算机 2006(z1)