## 中文文本预处理

* 1. 分词软件 调用 （中科院分词系统）
     1. 软件下载：<http://ictclas.nlpir.org/>
     2. 软件包目录&介绍

|   Readme.txt-------------------------->介绍

|

+---bin

|   +---DocExtractor----------->文档篇章语义抽取系统

|   |       DocExtractor.bat-->批处理，可以针对指定的文件夹进行语义抽取

|   |       DocExtractor.dll-->支撑的动态链接库，基于分词基础上

|   |       DocExtractorSample.exe-->应用程序

|   |

|   \---ICTCLAS2015----------->分词系统

|           ICTCLAS-tools.exe-->分词的支撑工具，可用于测试，本处主要用来做用户词典导入

|           importuserdict.bat-->可将用户词典自动导入到系统内

|           NLPIR.dll-->Win32下的支撑动态链接库，其他环境的库，可以访问lib对应环境的库文件

|           NLPIR.lib

|           NLPIR\_WinDemo.exe-->Win32下的演示程序，在Win8 32位下编译而成，部分环境可能不支持，或者显示异常

|           userdic.txt-->用户词典，用户可以自行编辑

|

+---Data-->系统核心词库

|   \---English-->英文处理的支持知识库，如果不需要英文处理的功能，可以不加载本库。

|

+---doc-->相关文档支持

|       ICTPOS3.0.doc-->我们的词性标注集说明

|       NLPIR-ICTCLAS2015分词系统开发手册.pdf-->开发使用手册

|

+---include-->系统头文件

|       NLPIR.h

|

+---lib-->不同环境下的支撑库，每一种库，同时支持C/C++/C#/Java库。其他小众化的环境支持，请联系我们

|   +---linux32-->Linux 32bit操作系统下的支持库

|   |       libNLPIR.so

|   |

|   +---linux64-->Linux 64bit操作系统下的支持库

|   |       libNLPIR.so

|   |       Readme.txt

|   |

|   +---win32-->Win 32bit操作系统下的支持库

|   |       NLPIR.dll

|   |       NLPIR.lib

|   |

|   \---win64-->Win 64bit操作系统下的支持库

|           NLPIR.dll

|           NLPIR.lib

|

+---ppt-->NLPIR2015开幕演讲的内容

|       NLPIR2014开幕演讲.pptx

|

+---sample-->NLPIR2015示例程序，均调试通过

|   +---C-->NLPIR2015 C语言示例程序

|   |       Example-C.cpp

|   |       Makefile: Linux下的编译makefile

|   +---C#-->NLPIR2014 C#语言示例程序

|   |       Example.cs

|   |

|   +---pythonsample-->NLPIR2015 Python示例程序

|   |

|   \---JnaTest\_NLPIR-->NLPIR2015 Java采用JNA调用的示例程序

|       |   .classpath

|       |   .project

|       |   nlpir.properties

|       |   Readme.txt

|       |

|       +---.settings

|       |       org.eclipse.core.resources.prefs

|       |       org.eclipse.jdt.core.prefs

|       |

|       +---bin

|       |   +---code

|       |   |       NlpirTest$CLibrary.class

|       |   |       NlpirTest.class

|       |   |

|       |   \---utils

|       |           ReadConfigUtil.class

|       |           SystemParas.class

|       |

|       +---lib

|       |       jna-4.0.0.jar

|       |

|       \---src

|           +---code

|           |       NlpirTest.java

|           |

|           \---utils

|                   ReadConfigUtil.java

|                   SystemParas.java

|

\---test-->NLPIR2015测试文档集合

|   18届三中全会.TXT

|   English.txt

|   中英文混杂示例.txt

|   屌丝，一个字头的诞生.TXT

|   用户词典.txt

|

\---docs-->NLPIR2015测试文档集合

0-两栖战车亮相.txt

1-辽宁阜新官员涉嫌淫乱事件举报者被刑拘.txt

2-辽宁一男子女厕用手机拍照被拘(图).txt

3-两人打甲流疫苗后死亡 另有15例较严重异常反应.txt

4-English.txt

* + 1. 软件使用（C++）

需要文件：include/NLPIR.h、lib文件夹下根据系统和版本选择不同的库文件、Data文件夹。

常用接口：

* NLPIR\_Init: *bool NLPIR\_Init(const char \* sInitDirPath=0,int encoding=GBK\_CODE,const char\*sLicenceCode=0)*

接口说明：软件初始化。

参数说明：sInitDirPath 初始化文件夹位置，表示Configure.xml和Data文件夹所在位置，默认值为0，表示当前目录；encoding 输入语料的编码格式，默认为GBK\_CODE，可选参数为UTF8\_CODE和BIG5\_CODE；sLicenseCode 证书编码，默认可不填。

* NLPIR\_Exit：bool NLPIR\_Exit()

接口说明：软件运行结束，释放内存。

* NLPIR\_ImportUserDict：unsigned int NLPIR\_ImportUserDict(const char \*sFilename,bool bOverwrite =true)

接口说明：引入用户自定义词典

参数说明：sFilename 用户词典文件名；bOverwrite 是否覆盖现有词典，默认为true，改为false将该词典添加到现有词典。

* NLPIR\_ParagraphProcess：const char \* NLPIR\_ParagraphProcess(const char \*sParagraph,int bPOStagged=1)

接口说明：分词函数

参数说明：sParagraph 需要分词的字符串；bPOStagged 分词结果是否包含POS标签，即词性标签，默认为true

* 其他接口函数请参考 Doc目录下开发手册
  + 1. 软件使用（Java）

主要通过Jna的方式对C++接口进行调用。

需要文件：lib下根据系统和版本选用不同的资源库文件、Data文件夹和jna.jar

使用方法：

Jna 编程首先根据 C 的头文件来声明对应的函数,声明后就像调用普通的 java 方法一样使用 即可,详细使用例子,请见代码【注意:我们的 dll 是通用的,C、java、C#所使用的 dll 是 同一个】。

样例代码：

NlpirTest 类就是对应的分词的 C 头文件的函数的声明:

public class NlpirTest {

// 定义接口CLibrary，继承自com.sun.jna.Library

public interface CLibrary extends Library {

// 定义并初始化接口的静态变量 这一个语句是来加载 dll 的,注意 dll 文件的路径 可以是绝对路径也可以是相对路径,只需要填写 dll 的文件名,不能加后缀。

CLibrary Instance = (CLibrary) Native.loadLibrary(

"E://java//JNI//JnaTest\_NLPIR//NLPIR", CLibrary.class);

// 初始化函数声明

public int NLPIR\_Init(byte[] sDataPath, int encoding,

byte[] sLicenceCode);

// 执行分词函数声明

public String NLPIR\_ParagraphProcess(String sSrc, int bPOSTagged);

// 提取关键词函数声明

public String NLPIR\_GetKeyWords(String sLine, int nMaxKeyLimit,

boolean bWeightOut);

// 退出函数声明

public void NLPIR\_Exit();

}

public static String transString(String aidString, String ori\_encoding,

String new\_encoding) {

try {

return new String(aidString.getBytes(ori\_encoding), new\_encoding);

} catch (UnsupportedEncodingException e) {

e.printStackTrace();

}

return null;

}

public static void main(String[] args) throws Exception {

String argu = "";

// String system\_charset = "GBK";//GBK----0

String system\_charset = "GBK";

int charset\_type = 1;

// int charset\_type = 0;

// 调用printf打印信息

int init\_flag = CLibrary.Instance.NLPIR\_Init(argu

.getBytes(system\_charset), charset\_type, "0"

.getBytes(system\_charset));

if (0 == init\_flag) {

System.err.println("初始化失败！");

return;

}

String sInput = "据悉，质检总局已将最新有关情况再次通报美方，要求美方加强对输华玉米的产地来源、运输及仓储等环节的管控措施，有效避免输华玉米被未经我国农业部安全评估并批准的转基因品系污染。";

String nativeBytes = null;

try {

nativeBytes = CLibrary.Instance.NLPIR\_ParagraphProcess(sInput, 1);

System.out.println("分词结果为： " + nativeBytes);

int nCountKey = 0;

String nativeByte = CLibrary.Instance.NLPIR\_GetKeyWords(sInput, 10,false);

System.out.print("关键词提取结果是：" + nativeByte);

CLibrary.Instance.NLPIR\_Exit();

} catch (Exception ex) {

// TODO Auto-generated catch block

ex.printStackTrace();

}

}

}

* 1. 句法分析软件调用（斯坦福）
     1. 软件下载：<http://nlp.stanford.edu/software/lex-parser.shtml>
     2. 软件说明

可以根据不同的需要下载不同语言的版本，这里仅介绍Java版本的使用方法。

解压文件中lexparser-gui.bat进行可视化页面运行，解析需要的模型文件存放在stanford-parser-3.3.0-models.jar，可以对其解压，方面以后使用。在中文处理方面，提供的模型文件有chineseFactored.ser.gz、chinesePCFG.ser.gz、xinhuaFactored.ser.gz、xinhuaFactoredSegmenting.ser.gz、xinhuaPCFG.ser.gz。factored包含词汇化信息，PCFG是更快更小的模板，xinhua据说是根据大陆的《新华日报》训练的语料，而chinese同时包含香港和台湾的语料，xinhuaFactoredSegmenting.ser.gz可以对未分词的句子进行句法解析。

* + 1. API调用流程
* 指定模型文件，指定一些参数信息。

String grammars = "edu/stanford/nlp/models/lexparser/chinesePCFG.ser.gz";

* 加载模型文件，初始化用于句法分析的类LexicalizedParser。

LexicalizedParser lp = LexicalizedParser.loadModel(grammars);

* 调用parse()方法进行解析。其中s为待处理语料，最好为分词之后的词串。

Tree t = lp.parse(s);

* 后续处理，根据自己的需要生成不同的处理格式。

// 依存关系抽取

ChineseGrammaticalStructure gs = new ChineseGrammaticalStructure(t);

Collection<TypedDependency> tdl = gs.typedDependenciesCollapsed();