NLPIR/ICTCLAS 分词系统开发文档

自然语言处理与信息检索共享平台

Natural Language Processing & Information Retrieval Sharing Platform

http://ICTCLAS.nlpir.org/

@ICTCLAS 张华平博士

2017-8

For the latest information about NLPIR, please visit Http://ICTCLAS.nlpir.org/

访问 http://ictclas.nlpir.org/(自然语言处理与信息检索共享平台)，您可以获取 NLPIR 系统的最新版本，并欢迎您关注张华平博士的新浪微博 @ICTCLAS 张华平博士 交流。

Document Information

Document ID	NLPIR-ICTCLAS-2017-WHITEPAPER	Version	V4.0
Security level	Public 公开	Status	Creation and first draft for comment
Author	张华平	Date	Aug 31, 2017
Publisher	/	Approved by	

Version History

Note：The first version is"v0.1". Each subsequent version will add 0.1 to the exiting version. The version number should be updated only when there are significant changes, for example, changes made to reflect reviews. The first figure in the version 1.x denotes current review status by. 1. x denotes review process has passed round 1 etc .Anyone who create, review or modify the document should describe his action.

Version	Author/Reviewer	Date	Description
V1.0	Kevin Zhang	2011-8-21	first complete draft for comment. ICTCLAS2010
V2.0	Kevin Zhang	2012-8-21	complete draft for comment.ICTCLAS2012
V3.0	Kevin Zhang	2012-12-19	complete draft for comment.ICTCLAS2013
V4.0	Kevin Zhang	2013-12-19	complete draft for comment.ICTCLAS2014
V5.0	Kevin Zhang	2014-8-3	complete draft for comment.ICTCLAS2014 add some functions.
V6.0	Kevin Zhang	2014-12-25	complete draft for comment.ICTCLAS2014 add some functions.
V6.1	Kevin Zhang	2015-2-1	complete draft for comment.ICTCLAS add some functions.
V7.0	Kevin Zhang	2017-8-31	complete draft for comment.ICTCLAS add some functions.

目录

NLPIR/ICTCLAS 分词系统开发文档 1
目录 4
1. NLPIR/ICTCLAS 分词系统简介 5
2. NLPIR/ICTCLAS 分词系统主要功能介绍 6
3. NLPIR/ICTCLAS 分词系统评测 11
3.1 NLPIR/ICTCLAS 在 973 评测中的测试结果 11
3.2 第一届国际分词大赛的评测结果 12
3.3 NLPIR/ICTCLAS 的评测结果 12
4. NLPIR/ICTCLAS 大事记： 13
5. 分词功能 C/C＋＋接口 15
5.1 NLPIR_Init 15
5.2 NLPIR_Exit 16
5.3 NLPIR_ParagraphProcess 17
5.4 NLPIR_GetLastErrorMsg 18
5.5 NLPIR_ParagraphProcessA 19
5.6 NLPIR_GetParagraphProcessAWordCount 21
5.7 NLPIR_ ParagraphProcessAW 24
5.8 NLPIR_FileProcess 24
5.9 NLPIR_ImportUserDict 25
5.10 NLPIR_AddUserWord 28
5.11 NLPIR_CleanUserWord() 29
5.12 NLPIR_SaveTheUsrDic 29
5.13 NLPIR_DelUsrWord 30
5.14 NLPIR_GetUniProb 32
5.15 NLPIR_IsWord 32
5.16 NLPIR_IsUserWord 33
5.17 NLPIR_GetWordPOS 33
5.18 NLPIR_SetPOSmap 34
5.19 NLPIR_FinerSegment 35
5.20 NLPIR_GetEngWordOrign 35
5.21 NLPIR_WordFreqStat 36
5.22 NLPIR_FileWordFreqStat 36
5.23 class CNLPIR 37
5.24 GetActiveInstance 38
5.25 NLPIR_FingerPrint 39
6. 分词功能 JNA 接口 40
6.1 jna 使用分词说明 40
6.2 jna 使用分词示例 40
7. hadoop 平台使用分词 41
7.1 hadoop 使用分词说明 41

7.2 hadoop 使用分词示例 41
8. C#接口说明 44
8.1 说明 44
8.2 接口示例 44
9. NLPIR/ICTCLAS 运行环境 46
9.1 支持的环境 46
9.2 Linux 如何调用 NLPIR 47
10. 常见问题（FAQ） 47
Q1: Linux 调用 NLPIR 的时候，链接不上库 47
Q2: NLPIR 系统初始化老是失败 47
Q3: NLPIR 系统是否支持多线程，没有显式的创建与销毁分词对象（句柄、上下文）的接口，故不支持多线程和多实例 48
Q4: 没有找到选择粗/细颗粒度的接口 48
Q5: 连续的空白符号是每个符号单独输出的，希望有合并输出的选项。 48
Q6: 支持在一个应用中，同时进行 GB18030 和 UTF-8 的分词 48
Q7: NLPIR/ICTCLAS 的 JNI 调用实现过程 49
11. 作者简介 49

1. NLPIR/ICTCLAS 分词系统简介

词法分析是自然语言处理的基础与关键。张华平博士在多年研究工作积累的基础上，研制出了 NLPIR 分词系统，主要功能包括中文分词；英文分词；词性标注；命名实体识别；新词识别；关键词提取；支持用户专业词典与微博分析。NLPIR 系统支持多种编码（GBK 编码、UTF8 编码、BIG5 编码）、多种操作系统（Windows, Linux， FreeBSD 等所有主流操作系统）、多种开发语言与平台（包括：C/C++/C#,Java,Python,Hadoop 等）。

NLPIR 分词系统前身为 2000 年发布的 ICTCLAS 词法分析系统，从 2009 年开始，为了和以前工作进行大的区隔，并推广 NLPIR 自然语言处理与信息检索共享平台，调整命名为 NLPIR 分词系统。张华平博士先后倾力打造十余年，内核升级十余次，先后获得了 2010 年钱伟长中文信息处理科学技术奖一等奖，2003 年国际 SIGHAN 分词大赛综合第一名，2002 年国内 973 评测综合第一名。全球用户突破 30 万，包括中国移动、华为、中搜、3721、NEC、中华商务网、硅谷动力、云南日报等企业，清华大学、新疆大学、华南理工、麻省大学等机构：同时，ICTCLAS 广泛地被《科学时报》、《人民日报》海外版、《科技日报》等多家媒体报道。您可以访问 Google 进一步了解 ICTCLAS 的应用情况。

我们提供各类二次开发接口，特别欢迎相关的科研人员、工程技术人员使用，并承诺非商用应用永久免费的共享策略。访问 http://ictclas.nlpir.org/(自然语言处理与信息检索共享平台)，您可以获取 NLPIR 系统的最新版本，并欢迎您关注张华平博士的新浪微博 @ICTCLAS 张华平博士 交流。

图 1：NLPIR/ICTCLAS 获得了钱伟长中文信息处理科学技术奖一等奖

2. NLPIR/ICTCLAS 分词系统主要功能介绍

1）中英文混合分词功能

自动对中文英文信息进行分词与词性标注功能，涵盖了中文分词、英文分词、词性标注、未登录词识别与用户词典等功能，如图所示

图 2：中英文混合分词展示

2）关键词提取功能

采用交叉信息熵的算法自动计算关键词，包括新词与已知词，下面是对十八届三中全会报告部分内容的关键词提取结果。

图 3：十八届三中全会报告的关键词提取结果

3）新词识别与自适应分词功能

从较长的文本内容中，基于信息交叉熵自动发现新特征语言，并自适应测试语料的语言概率分布模型，实现自适应分词。

图 4：自动识别“屌丝”等新词，并自动调整分词结果，实现自适应分词

4）用户专业词典功能

可以单条导入用户词典，也可以批量导入用户词典。如可以定“举报信 敏感点”，其中举报信是用户词，敏感点是用户自定义的词性标记。

图 5：判别用户定义词“举报信”，设置为自定义词性“敏感点”

5）微博分词功能

对博主 ID 进行 nr 标示，对转发的会话进行自动分割标示（标示为 ssession），URL 以及 Email 进行自动标引。

图 6：微博分词示例

3. NLPIR/ICTCLAS 分词系统评测

3.1 NLPIR/ICTCLAS 在 973 评测中的测试结果

2002 年 7 月 6 日，NLPIR/ICTCLAS 参加了国家 973 英汉机器翻译第二阶段的开放评测，测试结果如下：

领域	词数	SEG	TAG1	RTAG
体育	33,348	97.01%	86.77%	89.31%
国际	59,683	97.51%	88.55%	90.78%
文艺	20,524	96.40%	87.47%	90.59%
法制	14,668	98.44%	85.26%	86.59%
理论	55,225	98.12%	87.29%	88.91%
经济	24,765	97,80%	86.25%	88.16%
总计	208,213	97,58%	87.32%	89.42%

表 3. ICTCLAS 在 973 评测中的测试结果

说明：

1. 数据来源：国家 973 英汉机器翻译第二阶段评测的评测总结报告
2. 标注相对正确率 RTAG=TAG1/SEG*100%
3. 由于我们采取的词性标注集和 973 专家组的标注集有较大出入，所以词性标注的正确率不具可比性
4. 专家组的开放评测结果表明：基于 HHMM 的 ICTCLAS 能实际的解决汉语词法分析问题，和兄弟单位的类似系统对比，ICTCLAS 的分词结果表现出色。

3.2 第一届国际分词大赛的评测结果

为了比较和评价不同方法和系统的性能，第四十一届国际计算语言联合会(41st Annual Meeting of the Association for Computational Linguistics, 41th ACL)下设的汉语特别兴趣研究组(the ACL Special Interest Group on Chinese Language Processing, SIGHAN；www.sighan.org) 于 2003 年 4 月 22 日至 25 日举办了第一届国际汉语分词评测大赛(First International Chinese Word Segmentation Bakeoff)[28]。报名参赛的分别是来自于大陆、台湾、美国等 6 个国家和地区，共计 19 家研究机构，最终提交结果的是 12 家参赛队伍。

大赛采取大规模语料库测试，进行综合打分的方法，语料库和标准分别来自北京大学（简体版）、宾州树库（简体版）、香港城市大学（繁体版），台湾“中央院” （繁体版）。每家标准分两个任务(Track)：受限训练任务(Close Track)和非受限训练任务(Open Track)。

NLPIR/ICTCLAS 分别参加了简体的所有四项任务，和繁体的受限训练任务。其中在宾州树库受限训练任务中综合得分 0.881[28]，名列第一；北京大学受限训练任务中综合得分 0.951[28]，名列第一；北京大学受限训练任务中综合得分 0.953[28]，名列第二。值得注意的是，我们在短短的两天之内，采取 ICTCLAS 简体版的内核代码，将多层隐马模型推广到繁体分词当中，同样取得了 0.938[28]的综合得分。

3.3 NLPIR/ICTCLAS 的评测结果

我们利用了《人民日报》1998 年 1 月的新闻纯文本语料进行开放测试，ICTCLAS3.0 测试的精度与速度如下表所示：

	开放测试一	开放测试二	开放测试三
功能描述	分词	分词+命名实体与新词识别	分词+命名实体与新词识别+词性标注
测试文件大小	4,092,478 Bytes	4,092,478 Bytes	4,092,478 Bytes
时间(s)	4.094000	6.467561	9.094001

核心数据所占内存	5.5MB	7.2MB	8.9MB
速度	999.63 KB/s	632.77 KB/s	450.02 KB/s
精度	分词精度：96.56%	分词精度：98.13%	分词精度：98.13% 词性标注精度：94.63%

说明：

1. 测试机器配置：CPU: PIV3.0G；内存：512M；

2. 分词精度指的是正确切分的词数占正确结果总词数的百分比；词性标注精度指的是切分与词性标注均正确的词数占正确结果总词数的百分比。

3. 开放测试：指的是测试样本不属于训练样本集合，否则称为封闭测试；封闭测试相当于考试试题都出自于学习过的书本，这种测试并没有实质意义，而往往有一些商家故意混淆视听，以封闭测试来冒充开放测试，制造准确率 99.5%的噱头，实际上，通过机械记忆小样本的封闭测试取得 100%的精度不存在任何问题。这一点特别提请用户注意。

4. NLPIR/ICTCLAS 大事记：

- 2000 年 5 月，张华平进入中科院计算所刘群教授所领导的自然语言处理课题组，开始从事分词的研发，2000 年 8 月第一版研制成功并发布，并发表第一篇分词的论文。
- 2002 年 7 月，在 973 项目"图像、语音、自然语言理解与知识挖掘"专家组的评测中，在所有参评的系统中，评测得分最高。（分词正确率高达 97.58%，参赛单位包括北京大学，清华大学等）
- 2003 年 1 月 7 日，获得国家版权局授予的软件著作权登记证书，编号为软著登字 005178 号）
- 在 2003 年 4 月 22 日至 25 日， ICTCLAS 参加了第四十一届国际计算语言联合会(41st Annual Meeting of the Association for Computational Linguistics, 41th ACL)下设的汉语特别兴趣研究组(the ACL Special Interest Group on Chinese Language Processing, SIGHAN)举办的第一届国际汉语分词评测大赛[10]，在参加的六项比赛中，获得了两项第一名、一项第二名。（参赛单位来自于 6 个国家和地区的 12 个系统，包括微软，SYSTRAN, Pennsylvania 大学，Berkeley 大学，北京大学）
- 作为计算所的 15 项免费技术成果之一，被来自于国内外的约 30000 人次的下载使用。作为中文自然语言处理开放平台的自由软件，受到了广泛的欢迎和关注，在《科学时报》、新浪网、人民日报海外版、中新网、新华网、人民网均有报道[11,12,13,14,15]。我们提供的各种形式研究成果，在

学术界和产业界得到了广泛的应用，其中包括：3721、NEC 研究院、中华商务网、硅谷动力、云南日报等企业，新疆大学、清华大学、华南理工、麻省大学等研究机构。

- 2004 年 7 月，推出 ICTCLAS2.0；
- 2005 年 12 月，推出 ICTCLAS2.6；
- 2006 年 4 月，推出 ICTCLAS3.0，速度接近 1MB/s，精度 98.13%；
- 2008 年初，推出 ICTCLAS2008;开始按照年份作为版本序号；
- 2010 年初，张华平博士调任北京理工大学，推出 ICTCLAS2010，并将名称调整为 NLPIR。2010 年获得了钱伟长中文信息处理科学技术奖一等奖。
- 2012 年 11 月，推出 NLPIR/ICTCLAS2013，增加了自适应分词、新词识别与关键词提取功能。第一次采用社交网络的形式发布内测。将库文件的名称统一改为 libNLPIR.so/NLPIR.dll
- 2013 年 12 月，推出 NLPIR/ICTCLAS，第一次进行线下的分词用户交流大会。

5. 分词功能 C/C＋＋接口

5.1 NLPIR_Init

Init the analyzer and prepare necessary data for NLPIR according the configure file.
int NLPIR_Init(const char * sDataPath=0,int encode=GBK_CODE,const char*sLicenceCode=0);

Routine	Required Header
NLPIR_Init	<NLPIR.h>

Return Value

Return true if init succeed. Otherwise return false.

Parameters

const char * sInitDirPath=NULL
sDataPath: Path where Data directory stored. the default value is NULL, it indicates the initial directory is current working directory path

encode: encoding code; encoding of input string, default is GBK_CODE (GBK encoding), and it can be set with UTF8_CODE (UTF8 encoding) and BIG5_CODE (BIG5 encoding).

sLicenseCode: license code for unlimited usage. common user ignore it

Remarks

The **NLPIR_Init** function must be invoked before any operation with NLPIR. The whole system need call the function only once before starting NLPIR. When stopping the system and make no more operation, NLPIR_Exit should be invoked to destroy all working buffer. Any operation will fail if init do not succeed.

NLPIR_Init fails mainly because of two reasons: 1) Required data is incompatible or missing 2) Configure file missing or invalid parameters. Moreover, you could learn more from the log file NLPIR.log in the default directory.

Example

```
#include "NLPIR.h"
#include <stdio.h>
#include <string.h>

int main(int argc, char* argv[])
{
```



```
//Sample1: Sentence or paragraph lexical analysis with only one result
char sSentence[2000];

const char * sResult;
if(!NLPIR_Init("D:/NLPIR",UTF8_CODE))//数据在当前路径下，设置为 UTF8 编码的分词

{     printf("ICTCLAS INIT FAILED! Reason is %s\n", NLPIR_GetLastErrorMsg());

      return ;
}printf("Input sentence now('q' to quit)!\n");
scanf("%s",sSentence);
while(_stricmp(sSentence,"q")!=0)
{
sResult = NLPIR_ParagraphProcess(sSentence,0);
printf("%s\nInput string now('q' to quit)!\n", sResult);
scanf("%s",sSentence);
}
NLPIR_Exit();
return 0;
}
```

Output

5.2 NLPIR_Exit

Exit the program and free all resources and destroy all working buffer used in NLPIR.
bool NLPIR_Exit();

Routine	Required Header
NLPIR_Exit	<NLPIR.h>

Return Value

Return true if succeed. Otherwise return false.

Parameters

none

Remarks

The **NLPIR_Exit** function must be invoked while stopping the system and make no more operation. And call NLPIR_Init function to restart NLPIR.

Example


```
#include "NLPIR.h"
#include <stdio.h>
#include <string.h>

int main(int argc, char* argv[])
{
//Sample1: Sentence or paragraph lexical analysis with only one result
char sSentence[2000];

const char * sResult;
if(!NLPIR_Init())
{
printf("Init fails\n");
return -1;
}
printf("Input sentence now('q' to quit)!\n");
scanf("%s",sSentence);
while(_stricmp(sSentence,"q")!=0)
{
sResult = NLPIR_ParagraphProcess(sSentence,1);
printf("%s\nInput string now('q' to quit)!\n", sResult);
scanf("%s",sSentence);
}
NLPIR_Exit();
return 0;
}
```

Output

5.3 NLPIR_ParagraphProcess

Process a paragraph, and return the result buffer pointer
const char * NLPIR_ParagraphProcess(const char *sParagraph,int bPOStagged=1);

Routine	Required Header
NLPIR_ParagraphProcess	<NLPIR.h>

Return Value

Return the pointer of result buffer.

Parameters

sParagraph: The source paragraph

bPOStagged: Judge whether need POS tagging, 0 for no tag; 1 for tagging; default:1.

Remarks

The **NLPIR_ParagraphProcess** function works properly only if **NLPIR_Init** succeeds.

Example

```
#include "NLPIR.h"
#include <stdio.h>
#include <string.h>

int main(int argc, char* argv[])
{
//Sample1: Sentence or paragraph lexical analysis with only one result
char sSentence[2000];
const char *sResult;
if(!NLPIR_Init())
{
printf("Init fails\n");
return -1;
}
printf("Input sentence now('q' to quit)!\n");
scanf("%s",sSentence);
while(_stricmp(sSentence,"q")!=0)
{
sResult=NLPIR_ParagraphProcess(sSentence,1);
printf("%s\nInput string now('q' to quit)!\n",sResult);
scanf("%s",sSentence);
}
NLPIR_Exit();
return 0;
}
```

Output

5.4 NLPIR_GetLastErrorMsg

Get last error message from the system. It can help us find and debug the possible problems in NLPIR-ICTCLAS.

const char * NLPIR_GetLastErrorMsg();

Routine	Required Header
NLPIR_GetLastErrorMsg	<NLPIR.h>

Return Value

Return the error message.

Parameters

None

Remarks

Example:

```
Void main()
{
	if(!NLPIR_Init("D:/NLPIR",UTF8_CODE))//数据在当前路径下，设置为UTF8编码的分词
	{
		printf("ICTCLAS INIT FAILED! Reason is %s\n", NLPIR_GetLastErrorMsg());
		return ;
	}
	printf("ICTCLAS INIT Success! \n");
	NLPIR_Exit();
	return;
}
```

5.5 NLPIR_ParagraphProcessA

Process a paragraph

const result_t * NLPIR_ParagraphProcessA(const char *sParagraph,int *pResultCount,bool bUserDict=true);

Routine	Required Header
NLPIR_ParagraphProcessA	<NLPIR.h>

Return Value

the pointer of result vector, it is managed by system, user cannot alloc and free it

```
struct result_t{
   int start; //start position,词语在输入句子中的开始位置
   int length; //length,词语的长度
   char    sPOS[POS_SIZE];//word type，词性ID值，可以快速的获取词性表
   int       iPOS;//词性
   int word_ID; //如果是未登录词，设成或者-1
   int word_type; //区分用户词典;1，是用户词典中的词；，非用户词典中的词
   int weight;// word weight
  };
```


Parameters

sParagraph: The source paragraph
pResultCount: pointer to result vector size

bUserDict:whether use UserDict

Remarks

The **NLPIR_ParagraphProcessA** function works properly only if **NLPIR_Init** succeeds.

Example

```
#include "NLPIR.h"
#include <stdio.h>
#include <string.h>

int main(int argc, char* argv[])
{
//Sample1: Sentence or paragraph lexical analysis with only one result
char sSentence[2000];
const result_t *pVecResult;
int nCount;
if(!NLPIR_Init())
{
printf("Init fails\n");
return -1;
}
printf("Input sentence now!\n");
scanf("%s",sSentence);
while(_stricmp(sSentence,"q")!=0)
{
pVecResult=NLPIR_ParagraphProcessA(sInput,&nCount,true);
for (int i=0;i<nCount;i++)
{
printf("Start=%d Length=%d Word_ID=%d POS_ID=%d\n",
pVecResult[i].start,
pVecResult[i].length,
pVecResult[i].word_ID,
pVecResult[i].POS_id);
}
}
NLPIR_Exit();
return 0;
}
```


Output

5.6 NLPIR_GetParagraphProcessAWordCount

Get ProcessAWordCount, API for C#. Get word count and it helps us prepare the proper size buffer for result_t vector

int NLPIR_GetParagraphProcessAWordCount(const char *sParagraph);

Routine	Required Header
NLPIR_FileProcess	<NLPIR.h>

Return Value

result vector size

Parameters

sParagraph: The source paragraph

Remarks

The **NLPIR_GetParagraphProcessAWordCount** function works properly only if **NLPIR_Init** succeeds.
The output format is customized in NLPIR configure.

Example

```
using System;
using System.IO;
using System.Runtime.InteropServices;


namespace win_csharp
{
    [StructLayout(LayoutKind.Explicit)]
    public struct result_t
    {
        [FieldOffset(0)] public int start;
        [FieldOffset(4)] public int length;
        [FieldOffset(8)] public int sPos;
        [FieldOffset(12)] public int sPosLow;
        [FieldOffset(16)] public int POS_id;
        [FieldOffset(20)] public int word_ID;
        [FieldOffset(24)] public int word_type;
        [FieldOffset(28)] public int weight;
```



```
    }
    /// <summary>
    /// Class1 的摘要说明。
    /// </summary>
    class Class1
    {
        const string path = @"NLPIR30.dll";

        [DllImport(path,CharSet=CharSet.Ansi,EntryPoint="NLPIR_Init")]
        public static extern bool NLPIR_Init(String sInitDirPath);

        [DllImport(path,CharSet=CharSet.Ansi,EntryPoint="NLPIR_ParagraphProcess")]
        public static extern String NLPIR_ParagraphProcess(String sParagraph,int
bPOStagged);

        [DllImport(path,CharSet=CharSet.Ansi,EntryPoint="NLPIR_Exit")]
        public static extern bool NLPIR_Exit();

       [DllImport(path,CharSet=CharSet.Ansi,EntryPoint="NLPIR_ImportUserDict")]
        public static extern int NLPIR_ImportUserDict(String sFilename);

        [DllImport(path,CharSet=CharSet.Ansi,EntryPoint="NLPIR_FileProcess")]
        public static extern bool NLPIR_FileProcess(String sSrcFilename,String
sDestFilename,int bPOStagged);

        [DllImport(path,CharSet=CharSet.Ansi,EntryPoint="NLPIR_FileProcessEx")]
        public static extern bool NLPIR_FileProcessEx(String sSrcFilename,String
sDestFilename);

    [DllImport(path,CharSet=CharSet.Ansi,EntryPoint="NLPIR_GetParagraphProcessAWordCo
unt")]
        static extern int NLPIR_GetParagraphProcessAWordCount(String sParagraph);
        //NLPIR_GetParagraphProcessAWordCount
        [DllImport(path,CharSet=CharSet.Ansi,EntryPoint="NLPIR_ParagraphProcessAW")]
        static extern void NLPIR_ParagraphProcessAW(int nCount,
[Out,MarshalAs(UnmanagedType.LPArray)] result_t[] result);

        [DllImport(path, CharSet = CharSet.Ansi, EntryPoint = "NLPIR_AddUserWord")]
        static extern int NLPIR_AddUserWord(String sWord);

        [DllImport(path, CharSet = CharSet.Ansi, EntryPoint = "NLPIR_SaveTheUsrDic")]
        static extern int NLPIR_SaveTheUsrDic();
```



```
[DllImport(path, CharSet = CharSet.Ansi, EntryPoint = "NLPIR_DelUsrWord")]
static extern int NLPIR_DelUsrWord(String sWord);
/// <summary>
///  应用程序的主入口点。
/// </summary>
[STAThread]
static void Main(string[] args)
{
    //
    // TODO:  在此处添加代码以启动应用程序
    //
    if(!NLPIR_Init(null))
    {
        System.Console.WriteLine("Init NLPIR failed!");
        return;
    }

    String s ="点击下载超女纪敏佳深受观众喜爱。禽流感爆发在非典之后。";
    int count = NLPIR_GetParagraphProcessAWordCount(s);//先得到结果的词
数


    result_t[] result = new result_t[count];//在客户端申请资源

    NLPIR_ParagraphProcessAW(count,result);//获取结果存到客户的内存中

   int i=1;
    foreach(result_t r in result)
    {
        String sWhichDic="";
        switch (r.word_type)
        {
            case 0:
                sWhichDic = "核心词典";
                break;
            case 1:
                sWhichDic = "用户词典";
                break;
            case 2:
                sWhichDic = "专业词典";
                break;
            default:
                break;
```



```
                    }
                    Console.WriteLine("No.{0}:start:{1}, length:{2},POS_ID:{3},Word_ID:{4},
UserDefine:{5}, Word:{6}\n", i++, r.start, r.length, r.POS_id, r.word_ID, sWhichDic,
s.Substring(r.start / 2, r.length / 2));
                }

                 NLPIR_Exit();

            }
        }
}
```

Output

5.7 NLPIR_ ParagraphProcessAW

Process a paragraph, API for C#
void NLPIR_ParagraphProcessAW(int nCount,result_t * result);

Routine	Required Header
NLPIR_FileProcess	<NLPIR.h>

Return Value

None

Parameters

nCount: the paragraph word count.
result: Pointer to structure to store results.

Remarks

The **NLPIR_ParagraphProcessAW** function works properly only if **NLPIR_Init** succeeds.
The output format is customized in NLPIR configure.

Example

(见上 5.6 例子)

Output

5.8 NLPIR_FileProcess

Process a text file
Double NLPIR_FileProcess(const char *sSourceFilename,const char *sResultFilename,int bPOStagged=1);

Routine	Required Header
NLPIR_FileProcess	<NLPIR.h>

Return Value

Return the processing speed if processing succeed. Otherwise return false.

Parameters

sSourceFilename: The source file name to be analysized;
sResultFilename: The result file name to store the results.
bPOStagged: Judge whether need POS tagging, 0 for no tag; 1 for tagging; default:1.

Remarks

The **NLPIR_FileProcess** function works properly only if **NLPIR_Init** succeeds.

The output format is customized in NLPIR configure.

Example

```
#include "NLPIR.h"

int main(int argc, char* argv[])
{
//Sample2: File text lexical analysis

if(!NLPIR_Init())
{
printf("Init fails\n");
return -1;
}
printf("Input sentence now('q' to quit)!\n");
NLPIR_FileProcess("Test.txt","Test_result.txt",1);
NLPIR_Exit();
return 0;
}
```

Output

5.9 NLPIR_ImportUserDict

Import user-defined dictionary from a text file.
unsigned int NLPIR_ImportUserDict(const char *sFilename,bool bOverwrite=false);

Routine	Required Header
NLPIR_ImportUserDict	<NLPIR.h>

Return Value

The number of lexical entry imported successfully

Parameters

sFilename: Text filename for user dictionary

bOverwrite: true, overwrite the existing dictionary
false, add to the existing dictionary

Remarks

The **NLPIR_ImportUserDict** function works properly only if **NLPIR_Init** succeeds.

The text dictionary file foramt see User-defined Lexicon.

You only need to invoke the function while you want to make some change in your customized lexicon or first use the lexicon. After you import once and make no change again, NLPIR will load the lexicon automatically if you set UserDict "on" in the configure file. While you turn UserDict "off", user-defined lexicon would not be applied.

用户词典需要注意的事项还包括：

1. 如果用户词有空格，需要采用[]括起来，例如： [Bill Clinton] nrf
2. 如果需要该用户词作为文章的关键词输出，必须用户词性标注为：key,如：科学发展观 key
3. 如果将一个词是人名，同时又希望作为关键词输出，则需要标注为 key_nr，如 钟南山 key_nr
4. 如果将一个词是地名，同时又希望作为关键词输出，则需要标注为 key_ns，如 钓鱼岛 key_ns
5. 如果将一个词是机构名，同时又希望作为关键词输出，则需要标注为 key_nr，如 国安委 key_nt

Example

```
#include <string.h>

void SplitGBK(const char *sInput)
{//分词演示
    //初始化分词组件
    if(!NLPIR_Init("../..",))//数据在当前路径下，默认为 GBK 编码的分词
    {
```



```
        printf("ICTCLAS INIT FAILED!\n");
        return ;
    }

    char sSentence[5000]="Bill Clinton 是美国总统，好像没来过北京理工大学，喜欢吃小尾羊！";
    const char * sResult;
    //////////////////////////////////////////////////////////////////////////
    //下面开始测试用户词典功能
    //////////////////////////////////////////////////////////////////////////
    //导入用户词典前

    printf("未导入用户词典：\n");
    sResult = NLPIR_ParagraphProcess(sSentence, 1);
    printf("%s\n", sResult);

    //导入用户词典后
    printf("\n 导入用户词典后：\n");
    int nCount = NLPIR_ImportUserDict("userdic.txt");//userdic.txt 覆盖以前的用户词典,里面有用户词：小尾羊
    //保存用户词典
    printf("导入%d 个用户词。\n", nCount);
    sResult = NLPIR_ParagraphProcess(sSentence, 1);
    printf("%s\n", sResult);

    //导入第二个用户词典后
    printf("\n 导入用户词典后：\n");
    nCount = NLPIR_ImportUserDict("userdictgbk.txt",false);//userdictgbk.txt 补充以前的用户词典；里面有用户词：[Bill Clinton] nrf
    //保存用户词典
    printf("导入%d 个用户词。\n", nCount);
    sResult = NLPIR_ParagraphProcess(sSentence, 1);
    printf("%s\n", sResult);
    //释放分词组件资源
    NLPIR_Exit();
}
```

Output

[Bill Clinton]/nr 是/vshi 美国/nsf 总统/n ， /wd 好像/v 没/v 来/vf 过/uguo 北京理

工大学/nt ， /wd 喜欢/vi 吃/v 小尾羊/nvercat ！ /wt

5.10 NLPIR_AddUserWord

Add a word to the user dictionary.

int NLPIR_AddUserWord(const char *sWord);

Routine	Required Header
NLPIR_AddUserWord	<NLPIR.h>

Return Value

Return 1 if add succeed. Otherwise return 0.

Parameters

sWord:the word added.

Remarks

The **NLPIR_AddUserWord** function works properly only if **NLPIR_Init** succeeds.

Example

```
#include "NLPIR.h"
#include <stdio.h>
#include <string.h>

int main(int argc, char* argv[])
{
//Sample1: Sentence or paragraph lexical analysis with only one result
char sSentence[2000];

const char * sResult;
if(!NLPIR_Init())
{
printf("Init fails\n");
return -1;
}
```

NLPIR_AddUserWord(“爱思客 n”);//添加词：爱思客\t 词性。

其中“爱思客”为要添加的词，“n”为词的词性，”\t”为分隔符

```
printf("Input sentence now('q' to quit)!\n");
scanf("%s",sSentence);
while(_stricmp(sSentence,"q")!=0)
```



```
{
sResult = NLPIR_ParagraphProcess(sString,0);
printf("%s\nInput string now('q' to quit)!\n", sResult);
scanf("%s",sSentence);
}
NLPIR_Exit();
return 0;
}
```

Output

5.11 NLPIR_CleanUserWord()

Clean all temporary added user words
int NLPIR_CleanUserWord();

Routine	Required Header
NLPIR_CleanUserWord()	<NLPIR.h>

Return Value

Return 1 if clean succeed. Otherwise return 0.

Parameters

Remarks

The **NLPIR_CleanUserWord()** function works properly only if **NLPIR_Init** succeeds.

Example

Output

5.12 NLPIR_SaveTheUsrDic

Save the user dictionary to disk.
int NLPIR_SaveTheUsrDic();

Routine	Required Header
NLPIR_SaveTheUsrDic	<NLPIR.h>

Return Value

Return 1 if save succeed. Otherwise return 0.

Parameters

Remarks

The **NLPIR_SaveTheUsrDic** function works properly only if **NLPIR_Init** succeeds.

Example

```
#include "NLPIR.h"
#include <stdio.h>
#include <string.h>

int main(int argc, char* argv[])
{
//Sample1: Sentence or paragraph lexical analysis with only one result
char sSentence[2000];

const char * sResult;
if(!NLPIR_Init())
{
printf("Init fails\n");
return -1;
}

NLPIR_AddUserWord(“爱思客 n”);//你好\t 词性

NLPIR_SaveTheUsrDic();//保存用户词典
printf("Input sentence now('q' to quit)!\n");
scanf("%s",sSentence);
while(_stricmp(sSentence,"q")!=0)
{
sResult = NLPIR_ParagraphProcess(sString,0);
printf("%s\nInput string now('q' to quit)!\n", sResult);
scanf("%s",sSentence);
}
NLPIR_Exit();
return 0;
}
```

Output

5.13 NLPIR_DelUsrWord

Delete a word from the user dictionary.

int NLPIR_DelUsrWord(const char *sWord);

Routine	Required Header
NLPIR_DelUsrWord	<NLPIR.h>

Return Value

Return -1, the word not exist in the user dictionary; else, the handle of the word deleted

Parameters

sWord:the word to be delete.

Remarks

The **NLPIR_DelUsrWord** function works properly only if **NLPIR_Init** succeeds.

Example

```
#include "NLPIR.h"
#include <stdio.h>
#include <string.h>

int main(int argc, char* argv[])
{
//Sample1: Sentence or paragraph lexical analysis with only one result
char sSentence[2000];

const char * sResult;
if(!NLPIR_Init())
{
printf("Init fails\n");
return -1;
}

NLPIR_AddUserWord(“iThinker  n”);//你好\t 词性

NLPIR_AddUserWord(“爱思客  n”);

NLPIR_DelUsrWord(“iThinker”);//删除 iThinker

NLPIR_SaveTheUsrDic();//保存用户词典
printf("Input sentence now('q' to quit)!\n");
scanf("%s",sSentence);
while(_stricmp(sSentence,"q")!=0)
{
sResult = NLPIR_ParagraphProcess(sString,0);
printf("%s\nInput string now('q' to quit)!\n", sResult);
scanf("%s",sSentence);
}
NLPIR_Exit();
```



```
return 0;
}
```

Output

5.14 NLPIR_GetUniProb

Get Unigram Probability

double NLPIR_GetUniProb(const char *sWord);

Routine	Required Header
NLPIR_GetUniProb	<NLPIR.h>

Return Value

The unitary probability of a word.

Parameters

sWord: input word

Remarks

None

Example

5.15 NLPIR_IsWord

Judge whether the word is included in the core dictionary

int NLPIR_IsWord(const char *sWord);

Routine	Required Header
NLPIR_IsWord	<NLPIR.h>

Return Value

1: exists; 0: no exists

Parameters

sWord: input word

Remarks

None

Example

5.16 NLPIR_IsUserWord

Judge whether the word is included in the user-defined dictionary

int NLPIR_IsUserWord(const char *sWord);

Routine	Required Header
NLPIR_IsUserWord	<NLPIR.h>

Return Value

1: exists; 0: no exists

Parameters

sWord: input word

Remarks

None

Example

5.17 NLPIR_GetWordPOS

Get the word Part-Of-Speech information

const char * NLPIR_GetWordPOS(const char *sWord);

Routine	Required Header
NLPIR_GetWordPOS	<NLPIR.h>

Return Value

Success or fail

Parameters

sWord: input word

Remarks

None

Example

5.18 NLPIR_SetPOSmap

select which pos map will use.

int NLPIR_SetPOSmap(int nPOSmap);

Routine	Required Header
NLPIR_SetPOSmap	<NLPIR.h>

Return Value

Return 1 if excute succeed. Otherwise return 0.

Parameters

Parameters :nPOSmap : ICT_POS_MAP_FIRST 计算所一级标注集
ICT_POS_MAP_SECOND 计算所二级标注集
PKU_POS_MAP_SECOND 北大二级标注集
PKU_POS_MAP_FIRST 北大一级标注集

Remarks

The **NLPIR_SetPOSmap** function works properly only if **NLPIR_Init** succeeds.

Example

```
#include "NLPIR.h"
#include <stdio.h>
#include <string.h>

int main(int argc, char* argv[])
{
//Sample1: Sentence or paragraph lexical analysis with only one result
char sSentence[2000];

const char * sResult;
if(!NLPIR_Init())
{
printf("Init fails\n");
return -1;
}
```

NLPIR_SetPOSmap(ICT_POS_MAP_FIRST);

```
printf("Input sentence now('q' to quit)!\n");
scanf("%s",sSentence);
while(_stricmp(sSentence,"q")!=0)
```



```
{
sResult = NLPIR_ParagraphProcess(sString,0);
printf("%s\nInput string now('q' to quit)!\n", sResult);
scanf("%s",sSentence);
}
NLPIR_Exit();
return 0;
}
```

Output

5.19 NLPIR_FinerSegment

当前的切分结果过大时，如“中华人民共和国”需要执行该函数，将切分结果细分为“中华人民 共和国”细分粒度最大为三个汉字

const char* NLPIR_FinerSegment(const char *sLine);

Routine	Required Header
NLPIR_FinerSegment	<NLPIR.h>

Return Value

返回细粒度分词，如果不能细分，则返回为空字符串""

Parameters

sLine:输入的字符串

Remarks

None

Example

5.20 NLPIR_GetEngWordOrign

获取各类英文单词的原型，考虑了过去分词、单复数等情况

const char* NLPIR_GetEngWordOrign(const char *sWord);

Routine	Required Header
NLPIR_GetEngWordOrign	<NLPIR.h>

Return Value

返回的词原型形式

driven->drive drives->drive drove-->drive

Parameters

sWord:输入的单词

Remarks

None

Example

5.21 NLPIR_WordFreqStat

获取各类英文单词的原型，考虑了过去分词、单复数等情况

const char* NLPIR_WordFreqStat(const char *sText);

Routine	Required Header
NLPIR_WordFreqStat	<NLPIR.h>

Return Value

返回的是词频统计结果形式如下：

张华平/nr/10#博士/n/9#分词/n/8

Parameters

sWord:输入的文本内容

Remarks

None

Example

5.22 NLPIR_FileWordFreqStat

获取输入文本的词，词性，频统计结果，按照词频大小排序

const char* NLPIR_FileWordFreqStat(const char *sFilename);

Routine	Required Header
NLPIR_FileWordFreqStat	<NLPIR.h>

Return Value

返回的是词频统计结果形式如下：

张华平/nr/10#博士/n/9#分词/n/8

Parameters

sFilename 文本文件的全路径

Remarks

None

Example

5.23 class CNLPIR

NLPIR 类，使用之前必须调用NLPIR_Init(),退出必须调用NLPIR_Exit。在使用过程中可以使用多份CNLPIR，支持多线程分词处理。每个线程先调用GetActiveInstance，获取处理类，然后，设置SetAvailable(false)宣示线程主权，处理完成后，SetAvailable(true)释放线程主权

```
#ifdef OS_LINUX
class    CNLPIR {
#else
class    __declspec(dllexport) CNLPIR {
#endif
      public:
            CNLPIR();
            ~CNLPIR();
            double FileProcess(const char *sSourceFilename,const char *sResultFilename,int bPOStagged=1);
            //Process a file，类似于 C 下的 NLPIR_FileProcess
            const char * ParagraphProcess(const char *sLine,int bPOStagged=1);
            //Process a file，类似于 C 下的 NLPIR_ParagraphProcess
            const result_t * ParagraphProcessA(const char *sParagraph,int *pResultCount,bool bUserDict=true);
            //Process a file，类似于 C 下的 NLPIR_ParagraphProcessA

            void ParagraphProcessAW(int nCount,result_t * result);
            int GetParagraphProcessAWordCount(const char *sParagraph);

            bool SetAvailable(bool bAvailable=true);//当前线程释放该类，可为下一个线程使用
            bool IsAvailable();//判断当前分词器是否被线程占用
```



```
        unsigned int GetHandle()
        {
            return m_nHandle;
        }

#ifdef    NLPIR_KEY_NEW_FUNC//Include keyword and new word function
        const char * GetKeyWords(const char *sLine,int nMaxKeyLimit,bool bWeightOut);
        //获取关键词
        const char * GetFileKeyWords(const char *sFilename,int nMaxKeyLimit,bool
bWeightOut);
        //从文本文件中获取关键词
        const char * GetNewWords(const char *sFilename,int nMaxKeyLimit,bool
bWeightOut);
        //获取新词
        const char * GetFileNewWords(const char *sFilename,int nMaxKeyLimit,bool
bWeightOut);
        //从文本文件中获取新词
#endif
private:
        unsigned int m_nHandle;//该成员作为该类的 Handle 值，由系统自动分配，用户
不可修改
        bool m_bAvailable;//该成员作为多线程共享控制的参数，由系统自动分配，用
户不可修改
        int m_nThreadCount;//Thread Count
        bool m_bWriting;//writing    protection
};
```

5.24 GetActiveInstance

获取可用的CNLPIR类，适用于多线程开发，先获取可用的CNLP，再调用其中的功能

CNLPIR* GetActiveInstance();

Routine	Required Header
GetActiveInstance	<NLPIR.h>

Return Value

CNLPIR*

Parameters

None

Remarks

None

Example

5.25 NLPIR_FingerPrint

Extract a finger print from the paragraph .
unsigned long NLPIR_FingerPrint(const char *sLine);

Routine	Required Header
NLPIR_FingerPrint	<NLPIR.h>

Return Value

0, failed; else, the finger print of the content

Parameters

sLine：input text

Remarks

None

Example

```
#include "NLPIR.h"
#include <stdio.h>
#include <string.h>

int main(int argc, char* argv[])
{
//Sample1: Sentence or paragraph lexical analysis with only one result
char sSentence[2000];

if(!NLPIR_Init())
{
printf("Init fails\n");
return -1;
}

printf("Input sentence now('q' to quit)!\n");
scanf("%s",sSentence);
Int nCount = 0;
while(_stricmp(sSentence,"q")!=0)
```



```
{
unsigned long lFinger = NLPIR_FingerPrint(sString);
scanf("%s",sSentence);
}
NLPIR_Exit();
return 0;
}
```

Output

6. 分词功能 JNA 接口

6.1 jna 使用分词说明

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】。

6.2 jna 使用分词示例

..\NLPIR-ICTCLAS\projects\ICTCLAS_Java 找到该路径下的 java 项目。
..\ICTCLAS_Java\src\com\lingjoin\util\OSInfo.java **获取组件路径**
..\ICTCLAS_Java\src\com\lingjoin\demo\NlpirLib.java **分词组件的 Java 接口，采用 JNA 技术来实现**
..\ICTCLAS_Java\src\com\lingjoin\demo\NlpirMethod.java **分词组件方法类**
..\ICTCLAS_Java\src\com\lingjoin\test\NlpirTest.java **分词组件测试**

```
/**
     * 测试文本分词
     */
    @Test
    public void testParagraphProcess(){
        String content = "据俄罗斯卫星网 8 月 11 日发布美国《国家利益》杂志刊登的文章称，中国购买俄制苏-27 第四代战机，为本国空军翻开了现代史的页章。从那时起，中国空军日益强大。";
        String result = NlpirMethod.NLPIR_ParagraphProcess(content, 1);
        System.out.println(result);
    }
```

Output:
分词结果为： 据/p 俄罗斯/nsf 卫星网/user 8 月/t 11 日/t 发布/v 美国/nsf 《/wkz 国家/n 利益/n 》/wky 杂志/n 刊登/v 的/ude1 文章/n 称/v ，/wd 中国/ns 购买/v 俄/b 制/v 苏/b -/wp 27/m 第四/m 代/q 战机/n ，/wd 为/p 本国/rzs 空军/n 翻开/v 了/ule 现代史/n 的/ude1 页/q 章/n 。/wj 从/p 那时/rzt 起/f ，/wd 中国/ns 空军/n 日益/d 强大/a 。/wj

7. hadoop 平台使用分词

7.1 hadoop 使用分词说明

一个分布式系统基础架构，用户可以在不了解分布式底层细节的情况下，开发分布式程序。充分利用集群的威力高速运算和存储。在 hadoop 平台上使用分词的编程调用 dll 的方法不改变，依然使用 jna 的调用方式，只是实现的过程需要按照 hadoop 的编程要求来写，使用方式的示例请见代码【注意：我们的 dll 是通用的，C、java、C#所使用的 dll 是同一个】。

7.2 hadoop 使用分词示例

（1）Hadoop 使用分词，首先同样使用 jna 方式声明 C 的函数

```
import com.sun.jna.Library;

public abstract interface CLibrary extends Library {
    public abstract int NLPIR_Init(StringparamArrayOfByte1, int paramInt, StringparamArrayOfByte2);

    public abstract String NLPIR_ParagraphProcess(String paramString, int paramInt);

    public abstract String NLPIR_GetKeyWords(String paramString, int paramInt, boolean paramBoolean);

    public abstract void NLPIR_Exit();
}
```

（2）接着写 hadoop 的 job 类

```
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.input.TextInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
import org.apache.hadoop.util.GenericOptionsParser;


import java.io.IOException;
import java.net.URI;
import java.net.URISyntaxException;

public class WordSegmentationJob {
    public static void main(String args[]) throws IOException {
```



```
        Configuration conf = new Configuration();
        try {
            System.err.println(conf + "\n");
            String[] otherArgs = new GenericOptionsParser(conf, args).getRemainingArgs();
            if (otherArgs.length != 2) {
                System.err.println("Usage: word-seg <in> <out>");
                System.exit(2);
            }
            Utils.putClassFile("jna-4.0.0.jar", conf);
            Utils.putFile(new URI("data.zip"), conf);
            //NLPIR.init();
            //NLPIR.getInstance().NLPIR_Init("/work/nlpir".getBytes(), 1, "0".getBytes());

            Job job = new Job(conf, "word-seg");
            job.setJarByClass(WordSegmentationJob.class);
            //job.setNumReduceTasks(6);//0.97*(core*nodes)
            job.setMapperClass(WordSegmentationMapper.class);
            job.setInputFormatClass(TextInputFormat.class);
            job.setReducerClass(WordSegmentationReduce.class);
            FileInputFormat.addInputPath(job, new Path(otherArgs[0]));
            FileOutputFormat.setOutputPath(job, new Path(otherArgs[1]));
            System.exit(job.waitForCompletion(true) ? 0 : 1);
        } catch (IOException e) {
            e.printStackTrace();    //To change body of catch statement use File | Settings | File
Templates.
        } catch (InterruptedException e) {
            e.printStackTrace();    //To change body of catch statement use File | Settings | File
Templates.
        } catch (ClassNotFoundException e) {
            e.printStackTrace();    //To change body of catch statement use File | Settings | File
Templates.
        } catch (URISyntaxException e) {
            e.printStackTrace();    //To change body of catch statement use File | Settings | File
Templates.
        }
    }
}
```

（3）接着是 mapper 类的实现

```
import com.sun.jna.Native;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;

import java.io.File;
```



```
import java.io.IOException;

public class WordSegmentationMapper extends Mapper<LongWritable, Text, LongWritable,
Text> {
    private CLibrary cLibrary;

    @Override
    protected void setup(Context context) throws IOException, InterruptedException {
        System.err.println(new File(Utils.getPath("data",
context.getConfiguration()).toString()));
        Utils.unzipArchive(new File(Utils.getPath("data",
context.getConfiguration()).toString()));
        System.err.println(Utils.getRootPath() + "/libNLPIR.so");
        cLibrary = (CLibrary) Native.loadLibrary(Utils.getRootPath() + "/libNLPIR.so",
CLibrary.class);
        cLibrary.NLPIR_Init("/work/nlpir".getBytes(), 1, "0".getBytes());
    }

    @Override
    protected void cleanup(Context context) throws IOException, InterruptedException {
        super.cleanup(context);
        cLibrary.NLPIR_Exit();
        Utils.cleanFile();
    }

    @Override
    protected void map(LongWritable key, Text value, Context context) throws IOException,
InterruptedException {
        try {
            String nativeBytes =
NLPIR.getInstance().NLPIR_ParagraphProcess(value.toString(), 3);
            context.write(key, new Text(nativeBytes));
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

（4）最后实现主类

```
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Reducer;

import java.io.IOException;
import java.util.*;
```



```
public class WordSegmentationReduce extends Reducer<LongWritable, Text, Text, Text> {
    @Override
    protected void setup(Context context) throws IOException, InterruptedException {
    }

    @Override
    protected void reduce(LongWritable offset, Iterable<Text> values, Context context) throws
IOException, InterruptedException {
        Iterator<Text> iterator = values.iterator();
        if (iterator.hasNext()) {
            context.write(iterator.next(), new Text());
        }
    }

}
```

Output

分词结果为： 据悉/v ，/wd 质检/vn 总局/n 已/d 将/d 最新/a 有关/vn 情况/n 再次/d 通报/v 美方/n ，/wd 要求/v 美方/n 加强/v 对/p 输/v 华/b 玉米/n 的/ude1 产地/n 来源/n 、/wn 运输/vn 及/cc 仓储/vn 等/udeng 环节/n 的/ude1 管/v 控/v 措施/n ， /wd 有效/ad 避免/v 输/v 华/b 玉米/n 被/pbei 未经/d 我国/n 农业部/nt 安全/an 评估/vn 并/cc 批准/v 的/ude1 转基因/n 品系/n 污染/vn 。/wj

关键词提取结果是：美方#

关于 hadoop 的详细开发大家可以再网上查找相关资料，在这里只是贴上相关代码，不对 hadoop 的开发做详细的介绍。

8. C#接口说明

8.1 说明

C#调用c语言的dll方法很简单，声明调用的dll的方法即可，详细使用方法请见示例代码【注意：我们的dll是通用的，C、java、C#所使用的dll是同一个】。

8.2 接口示例

```
 const string path = @"NLPIR.dll";
        [DllImport(path, CharSet = CharSet.Ansi, CallingConvention =
CallingConvention.Winapi, EntryPoint = "NLPIR_Init")]
        public static extern bool NLPIR_Init(String sInitDirPath,int encoding,String
sLicenseCode);

NLPIR_ParagraphProcess(const char *sParagraph,int bPOStagged=1);
```



```
[DllImport(path, CharSet = CharSet.Ansi, CallingConvention = CallingConvention.Winapi, EntryPoint = "NLPIR_ParagraphProcess")]
        public static extern IntPtr NLPIR_ParagraphProcess(String sParagraph, int bPOStagged = 1);

        [DllImport(path,CharSet=CharSet.Ansi,EntryPoint="NLPIR_Exit")]
        public static extern bool NLPIR_Exit();

        [DllImport(path,CharSet=CharSet.Ansi,EntryPoint="NLPIR_ImportUserDict")]
        public static extern int NLPIR_ImportUserDict(String sFilename);

        [DllImport(path,CharSet=CharSet.Ansi,EntryPoint="NLPIR_FileProcess")]
        public static extern bool NLPIR_FileProcess(String sSrcFilename,String sDestFilename,int bPOStagged=1);

        [DllImport(path,CharSet=CharSet.Ansi,EntryPoint="NLPIR_FileProcessEx")]
        public static extern bool NLPIR_FileProcessEx(String sSrcFilename,String sDestFilename);


    [DllImport(path,CharSet=CharSet.Ansi,EntryPoint="NLPIR_GetParagraphProcessAWordCount")]
        static extern int NLPIR_GetParagraphProcessAWordCount(String sParagraph);
        //NLPIR_GetParagraphProcessAWordCount
        [DllImport(path,CharSet=CharSet.Ansi,EntryPoint="NLPIR_ParagraphProcessAW")]
        static extern void NLPIR_ParagraphProcessAW(int nCount, [Out,MarshalAs(UnmanagedType.LPArray)] result_t[] result);

        [DllImport(path, CharSet = CharSet.Ansi, EntryPoint = "NLPIR_AddUserWord")]
        static extern int NLPIR_AddUserWord(String sWord);

        [DllImport(path, CharSet = CharSet.Ansi, EntryPoint = "NLPIR_SaveTheUsrDic")]
        static extern int NLPIR_SaveTheUsrDic();

        [DllImport(path, CharSet = CharSet.Ansi, EntryPoint = "NLPIR_DelUsrWord")]
        static extern int NLPIR_DelUsrWord(String sWord);

       [DllImport(path, CharSet = CharSet.Ansi, EntryPoint = "NLPIR_NWI_Start")]
        static extern bool NLPIR_NWI_Start();

       [DllImport(path, CharSet = CharSet.Ansi, EntryPoint = "NLPIR_NWI_Complete")]
        static extern bool NLPIR_NWI_Complete();

       [DllImport(path, CharSet = CharSet.Ansi, EntryPoint = "NLPIR_NWI_AddFile")]
```



```
        static extern bool NLPIR_NWI_AddFile(String sText);

      [DllImport(path, CharSet = CharSet.Ansi, EntryPoint = "NLPIR_NWI_AddMem")]
        static extern bool NLPIR_NWI_AddMem(String sText);

      [DllImport(path, CharSet = CharSet.Ansi, CallingConvention =
CallingConvention.Winapi, EntryPoint = "NLPIR_NWI_GetResult")]
      public static extern IntPtr NLPIR_NWI_GetResult(bool bWeightOut = false);

     [DllImport(path, CharSet = CharSet.Ansi, EntryPoint = "NLPIR_NWI_Result2UserDict")]
        static extern uint NLPIR_NWI_Result2UserDict();

       [DllImport(path, CharSet = CharSet.Ansi,CallingConvention =
CallingConvention.Winapi, EntryPoint = "NLPIR_GetKeyWords")]
       public static extern IntPtr NLPIR_GetKeyWords(String sText,int nMaxKeyLimit=50,bool
bWeightOut=false);

       [DllImport(path, CharSet = CharSet.Ansi, CallingConvention =
CallingConvention.Winapi, EntryPoint = "NLPIR_GetFileKeyWords")]
       public static extern IntPtr NLPIR_GetFileKeyWords(String sFilename, int
nMaxKeyLimit = 50, bool bWeightOut = false);
```

说明：

1. 因为 C# 内存管理机制与 C 的差别，原来在 C 下面的函数 **const char * NLPIR_ParagraphProcess(const char *sParagraph,int bPOStagged=1); 在 C#下直接使用将导致内存出错，应当换成** public static extern int NLPIR_ParagraphProcessE(String sParagraph, StringBuilder sResult,int bPOStagged); 示例如下：

```
StringBuilder sResult = new StringBuilder(600);
NLPIR_ParagraphProcessE(s,sResult, 1);
```

9. NLPIR/ICTCLAS 运行环境

9.1 支持的环境

1. 可以支持 Windows、Linux、FreeBSD 等多种环境，支持普通 PC 机器即可运行。

2. 支持 GBK/UTF-8/BIG5

9.2 Linux 如何调用 NLPIR

1）与 window 下一样编程；
2）Makefile 的命令如下：
test: ../../../Src/ICTCLAS2013/example-c/Example-C.cpp ../../../Src/ICTCLAS2013/include/NLPIR.h
g++ ../../../Src/ICTCLAS2013/example-c/Example-C.cpp -L. -lpthread -L../../../bin/ICTCLAS2013 -lNLPIR -Wall -Wunused -O3 -DOS_LINUX -o ../../../bin/ICTCLAS2013/example
其中 Example-C.cpp 是测试使用 NLPIR 的程序；因为 NLPIR 进行了多线程的安全保护设计，需要调用多线程的库，即-L. –lpthread。调用 nlpir 的部分为：-L../../../bin/ICTCLAS2013 –lNLPIR 第一部分为路径，后面为 libNLPIR.so 对应的名称-lNLPIR。具体可以参见

10. 常见问题（FAQ）

常见的问题可以在线访问：ictclas.nlpir.org

Q1: Linux 调用 NLPIR 的时候，链接不上库

例如执行示例程序结果如下：
[root@localhost linux_c_sample]# ./test
./test: error while loading shared libraries: libNLPIR2011.so: cannot open shared object file: No such file or directory

Answer:

应当设置系统的 LD_LIBRARY_PATH，即：export LD_LIBRARY_PATH=./

Q2: NLPIR 系统初始化老是失败

Answer:

初始化失败一般原因包括：
1）系统在当前路径下，找不到系统配置文件 Configure.xml；
2）根据 Configure.xml，系统找不到指定的数据文件目录 data
3）Data 文件夹下面的文件缺失；
4）License 过期或者被封锁。
一般请查看当前目录下的 log 日志，一般名称为当前日期.log，其中有详细的介绍。

Q3: NLPIR 系统是否支持多线程，没有显式的创建与销毁分词对象（句柄、上下文）的接口，故不支持多线程和多实例

Answer:

支持多线程，全局初始化后，每个线程 new CNLPIR，即可在每个线程里面分词处理。

Q4: 没有找到选择粗/细颗粒度的接口

Answer:

请将 Configure.xml 中的参数设置为粗粒度。

<GranularityContorl>off</GranularityContorl> on 粗粒度，off 细粒度

Q5: 连续的空白符号是每个符号单独输出的，希望有合并输出的选项。

Answer:

分词的时候都是这么要求的，建议你可以考虑 CNLPIR 类中的

const result_t * ParagraphProcessA(const char *sParagraph,int *pResultCount);

里面只保存了分词结果，没有空格。需要的话，也可以合并。

Q6: 支持在一个应用中，同时进行 GB18030 和 UTF-8 的分词

Answer:

可以支持，但是一般都是建议编码标准化再分词，否则后续的应用很麻烦，我们有快速的编码转换程序。

Q7: NLPIR/ICTCLAS 的 JNI 调用实现过程

Answer:

参见天外的星星：http://blog.sina.com.cn/s/blog_5dc8d9a50100kwvj.html

11. 作者简介

张华平 博士 副教授 研究生导师
大数据搜索与挖掘实验室（北京市海量语言信息处理与云计算应用工程技术研究中心） 主任
地址：北京海淀区中关村南大街 5 号 100081
电话：+86-10-68918642 13681251543(助手电话)
Email:kevinzhang@bit.edu.cn
MSN: pipy_zhang@msn.com;
网站: http://www.nlpir.org (自然语言处理与信息检索共享平台)
http://www.bigdataBBS.com (大数据论坛)
微博:http://www.weibo.com/drkevinzhang/
微信公众号：大数据千人会
Dr. Kevin Zhang (张华平，Zhang Hua-Ping)
Associate Professor, Graduate Supervisor
Director, Big Data Search and Mining Lab.
Beijing Engineering Research Center of Massive Language Information Processing and Cloud Computing Application
Beijing Institute of Technology
Add: No.5, South St.,Zhongguancun,Haidian District,Beijing,P.R.C PC:100081
Tel: +86-10-68918642 13681251543(Assistant)

Email:kevinzhang@bit.edu.cn
MSN: pipy_zhang@msn.com;
Website: http://www.nlpir.org (Natural Language Processing and Information Retrieval Sharing Platform)
http://www.bigdataBBS.com (Big Data Forum)
Twitter: http://www.weibo.com/drkevinzhang/

Subscriptions: Thousands of Big Data Experts

自然语言处理与信息检索共享平台
Natural Language Processing & Information Retrieval Sharing Platform

