开发文档

http://9Eye.nlpir.org/

@ICTCLAS 张华平博士

2017-7

For the latest information about NLPIR, please visit http://9Eye.nlpir.org/

访问 http://9Eye.nlpir.org/ (九眼智能过滤)，您可以获取九眼智能过滤系统的最新版本，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-9EYE-2013-WHITEPAPER	Version	V4.0
Security level	Public 公开	Status	Creation and first draft for comment
Author	张华平	Date	Dec 19, 2013
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	2017-7-10	first complete draft for comment. 9Eye

目录

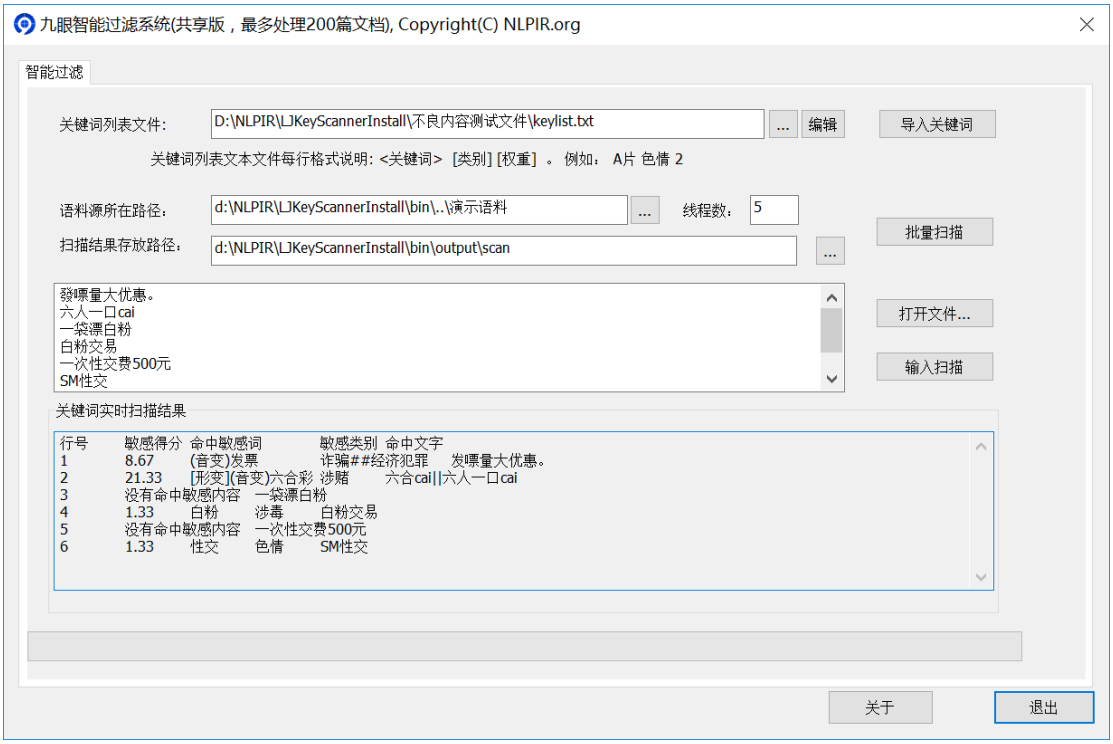
1. 九眼智能过滤系统简介

信息爆炸的时代也是信息过剩的时代，我们越来越感觉到被纷繁的信息所累，诈骗、传销、暴恐、色情、邪教胁迫、网络赌博、反伦理、假发票、语言暴力、垃圾广告等不良信息无孔不入，呼啸而来，我们深陷其中，甚至被裹挟吞噬。在这场信息与噪音的对抗中，我们如何冲出迷障，把握主动权？

显然，面对如此海量的信息流，传统人工过滤信息技术发挥的作用只能是杯水车薪，部分技术手段只能通过简单的关键词匹配，错漏百出。北理工大数据搜索与挖掘实验室结合多年的实战经验，利用多项自然语言处理专利算法，推出了九眼智能过滤系统，为不良信息的智能实时过滤提供了全新的技术解决方案！

九眼智能过滤系统面向复杂文本大数据的内容智能过滤系统，可实时智能识别不良关键词音变、形变与拆字等常见变体，并实现了语义的精准排歧，系统内置了国内最新最全的不良词库，适用于诈骗、传销、暴恐、色情、邪教胁迫、网络赌博、反伦理、假发票、语言暴力、垃圾广告等不良内容的智能过滤发现。

单机版演示界面如下：



信息安全问题关乎每个人，我们提倡数据共享，但我们更警惕信息安全！我们秉持着高度的社会责任感，致力于营造一个绿色健康的信息交流空间。我们相信:唯有真，可期待！

九眼智能过滤系统开创信息过滤新技术，将自然语言和人工智能紧密结合，打破传统局限，变被动为主动，将信息过滤工作智能化、语义化、快速化，探索信息监测新可能。

九眼智能过滤，管他千变万化！欢迎访问： http://9Eye.nlpir.org/ 使用。

2. 九眼智能过滤系统技术架构

整个系统包括敏感知识库脱机生成与在线并行核查两个过程。其中核心的采用了 专利算法“完美双数组 TRIE 树词典管理与检索方法（专利号：200510130690.3）。本发明涉及自然语言处理领域和信息检索领域，特别是一种完美双数组 TRIE 树词典管理与检索方法。将 Trie 树结构转换成两个线性数组表示，并在构造过程中提出了一种优化策略。同时提出一种自适应编码方案，以字节为编码单位对字符进行自动编码。包括步骤：(1)将词典用 Trie 树结构表示；(2)将 Trie 树转换成两个线性数组；(3)根据用户的输入，利用生成的两个数组实现对词典的检索。其具体步骤包括：步骤 1，对词典以字节为单位进行自动编码生成序列码文件；步骤 2，将词典用 Trie 树表示，利用序列码文件将 Trie 树转换成两个数组来表示；步骤 3，在生成的两个线性数组中检索用户提交的词。

3. 九眼智能过滤系统技术特色

九眼智能过滤三大技术特色：智能变种、语义排歧、快速实时

1、智能变种识别

利用美双数组 TRIE 树词典管理与检索方法，系统自动识别形变词、音变词、拆字、噪音、繁简体、全角半角、中间加各类干扰噪音等变体；如“發嘌”、“Fa 票”转换为同音词“卖发票”，将拆字“弓长”转换为“张”，“六人一口 cai”识别为“六合彩”。同时，系统支持自定义不良词库，增量添加百万量级词库。

变种自动识别：音变词、形变词、噪音、繁简体、全角半角等变体； 音变词：自动将不良词转换成全拼、简拼，如“發嘌”、“Fa 票”转换为同音词“卖发票”，将拆字“弓长”转换为“张”

2、语义排歧

九眼智能过滤利用 NLPIR 语义精准分词系统与情感分析系统，精准识别与过滤，排除正面无害的信息，极大降低了误判率。语义排歧示例：“一次[性交]费 5000 元”、“我[家宝]贝”、“[插入]银行卡”，买了一袋漂[白粉]。

3、快速实时

专利算法，快速扫描，单机速度 30MB/s；支持单机多线程、多机并行、Hadoop 云服务模式，对 PB 级不良内容实现并行高效在线核查。

九眼智能过滤特色与优势

九眼智能过滤系统是一套充分融合了自然语言理解、人工智能、大数据分析等领域尖端技术，具有智能、高效、自学习三大特点，其特色与优势无可比拟：

4. 二次开发 API 接口(C/C++版本)

4.1 KS_Init

Init the analyzer and prepare necessary data for NLPIR according the configure file.

```
KEYSCANAPI_API int KS_Init(const char *sInitDirPath = "", int encode = GBK_CODE, const char*sLicenceCode = 0,const char *sDelimiter=",");
```

Routine	Required Header
KS_Init	<KeyScanAPI.h>

Return Value

Return true if init succeed. Otherwise return false.

Parameters

sInitDirPath: Initial Directory Path, where file Configure.xml and Data directory stored. the default value is 0, it indicates the initial directory is current working directory path

int encoding: 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).

const char* sLicenceCode: license code, special use for some commercial users. Other users ignore the argument

const char* sDelimiter: Result delimiter using to separate deferent fields (score, rule, class and hit result), default as ",". For example "21.33 [形变](音变)六合彩 涉赌 六合 cai||六人一口 cai "

Remarks

The **KS_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, KS_Exit should be invoked to destroy all working buffer. Any operation will fail if init do not succeed.

KS_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 [Current Date].log in the default directory. Or call KS_GetLastErrorMsg() to get the reason.

Example

```
int main(int argc, char* argv[])
{
    //Usage
    if (argc<5)
    {
        printf("Usage1-import User Dictionary: %s i <dict_path>
<encoding:0-GBK,1-UTF8,2-BIG5,3-GBK+Fanti> <dictioanry_text_file>\n",argv[0]);
        printf("Usage1-Testing: %s t <dict_path>
<encoding:0-GBK,1-UTF8,2-BIG5,3-GBK+Fanti> <testing_text_file>\n",argv[0]);
        return -2;
    }
    if (!KS_Init(argv[2], atoi(argv[3])))
    {//Init Failed!
        printf("KS_Init Failed! Reason is %s\n", KS_GetLastErrorMsg());
        return -1;

    }

    KS_Exit();

}
```

Output

4.2 KS_Exit

Exit the program and free all resources and destroy all working buffer used in NLPIR.

```
bool KS_Exit();
```

Routine	Required Header
KS_Exit	<KeyScanAPI.h>

Return Value

Return true if succeed. Otherwise return false.

Parameters

none

Remarks

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

Example

See 4.1

Output

4.2 KS_NewInstance

Routine	Required Header
KS_NewInstance	<KeyScanAPI.h>

Return Value

Return KS Handle if succeed. Otherwise return false.

Parameters

none

Remarks

New a KeyScan Instance

The function must be invoked before mulitiple keyword scan filter

4.3 KS_DeleteInstance

```
/*************************************************************************
*
*  Func Name  : KS_DeleteInstance
*
*  Description: Delete a  KeyScan Instance with handle
*               The function must be invoked before release a specific classifier
*
*  Parameters : DC_HANDLE , KeyScan Handle
*  Returns    :
*  Author     : Kevin Zhang
*  History    :
*              1.create 2015-9-22
*************************************************************************/
KEYSCANAPI_API int KS_DeleteInstance(KS_HANDLE handle);
```


4.4 KS_ImportUserDict

Import user-defined dictionary from a text file.

```
unsigned int KS_ImportUserDict(const char *sFilename,bool bOverwrite
=true);
```

Routine	Required Header
KS_ImportUserDict	<KeyScanAPI.h>

Return Value

The number of lexical entry imported successfully

Parameters

sFilename: Text filename for user dictionary

bOverwrite: true(default), overwrite the existing dictionary
false, add to the existing dictionary

Remarks

The **KS_ImportUserDict** function works properly only if **KS_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.

关键词列表的格式，采用文本文件，每行的格式如下：

词条　　词类　权重　是否变形处理

例如： AV 电影 色情 2 0

六合彩 涉赌 1 1

Example

4.5 KS_Scan

```
/*************************************************************************
 *
 *  Func Name  : KS_Scan
 *
 *  Description: 扫描输入的文本内容，输出扫描结果
 *  Parameters : sContent:文本内容
 *               KS_HANDLE handle:  handle of KeyScanner
 *  Returns    : const char*: 涉及不良的所有类别与权重，按照权重排序。
 *                 如:  色情/10#暴力/1#
 *                 "": 表示无扫描命中结果
 *  Author     : Kevin Zhang
 *  History    :
 *              1.create 2014-8-3
 *  例如输出格式为:  政治反动/2#FLG/1#涉领导人/1#
*************************************************************************/
KEYSCANAPI_API const char* KS_Scan(const char*sContent, KS_HANDLE handle = 0);
```

4.6 KS_ScanDetail

```
/*************************************************************************
 *
 *  Func Name  : KS_ScanDetail
 *
 *  Description: 扫描输入的文本内容，输出命中的详细信息
 *  Parameters : sContent:文本内容
 *               KS_HANDLE handle:  handle of KeyScanner
 *  Returns    : 返回具体涉及不良的内容，并标引出来
 *  Author     : Kevin Zhang
 *  History    :
 *              1.create 2014-8-3
// 返回的格式如下:
// 返回值：返回包含了扫描结果的内容，扫描结果明细:
//       <class name="FLG" weight=1>法*輪大法</class>好!
*************************************************************************/
KEYSCANAPI_API const char* KS_ScanDetail(const char*sContent, KS_HANDLE handle = 0);
```

4.7 KS_ScanFile

```
/*************************************************************************
*
*  Func Name  : KS_ScanFile
*
```



```
*  Description: 扫描输入的文本文件内容
*  Parameters : sFilename:文本文件名
*                KS_HANDLE handle:  handle of KeyScanner
*  Returns    : const char*: 涉及不良的所有类别与权重，按照权重排序。
*                  如:  色情/10##暴力/1##
*                  "": 表示无扫描命中结果
*  Author     : Kevin Zhang
*  History    :
*              1.create 2014-8-3
*  文本文件的格式为:   词条    词类
*        例如:  AV电影 色情
**************************************************************************/
KEYSCANAPI_API const char* KS_ScanFile(const char    *sFilename, KS_HANDLE handle = 0);
```

4.7 KS_ScanFileDetail

```
/**************************************************************************
*
*  Func Name  : KS_ScanFileDetail
*
*  Description: 扫描输入的文本文件内容
*  Parameters : sFilename:文本文件名
*                KS_HANDLE handle:  handle of KeyScanner
*  Returns    : 返回具体涉及不良的内容，并标引出来
*  Author     : Kevin Zhang
*  History    :
*              1.create 2014-8-3
// 返回的格式如下:
// 返回值: 返回包含了扫描结果的内容，扫描结果明细:
//          *<class name="政治反动" weight=1>习*-包（）子</class>*(散财童子，真心不爽啊。
//       <class name="FLG" weight=1>法*輪大法</class>好!
**************************************************************************/
KEYSCANAPI_API const char* KS_ScanFileDetail(const char  *sFilename, KS_HANDLE handle =
0);
```

4.8 KS_ScanLine

```
/**************************************************************************
*
*  Func Name  : KS_ScanLine
*
*  Description: 按行扫描输入的文本文件内容
*  Parameters : sFilename:输入的文本文件名
*                sResultFilename: 输出的结果文件名
*                KS_HANDLE handle:  handle of KeyScanner
```



```
*                int bEncript:0 不加密；1，加密
*  Returns     : 返回涉及不良的内容行数
*  Author      : Kevin Zhang
*  History     :
*              1.create 2017-1-12
// 返回的格式如下：
// 返回值：返回包含了扫描结果的行号与内容，扫描结果明细:
//          *<class name="政治反动" weight=1>习*-包（）子</class>*(散财童子，真心不爽啊。
//       <class name="FLG" weight=1>法*輪大法</class>好!
*************************************************************************/
KEYSCANAPI_API int KS_ScanLine(const char   *sFilename, const char*sResultFilename,
KS_HANDLE handle = 0, int bEncript = false);
```

4.9 KS_ScanStat

```
/*************************************************************************
*
*  Func Name  : KS_ScanStat
*
*  Description: 输出扫描结果的命中统计报告，利于进一步的分析核查
*  Parameters : sResultFile: 输出结果的文件文件
*
*                KS_HANDLE handle:  handle of KeyScanner
*  Returns     : 成功扫描到问题的文件数
*  Author      : Kevin Zhang
*  History     :
*              1.create 2017-1-12
// 返回的格式如下：
*************************************************************************/
KEYSCANAPI_API int KS_ScanStat(const char *sResultFile, KS_HANDLE handle = 0);
```

4.9 KS_ScanDir

```
/*************************************************************************
*
*  Func Name  : KS_ScanDir
*
*  Description: 多线程扫描按行扫描输入的文本夹文件内容
*  Parameters : sInputDirPath:输入的文件夹路径
*               sResultPath: 输出结果的文件夹路径
*                nThreadCount: 线程数，默认10个
*                int bEncript:0 不加密；1，加密
*  Returns     : 成功扫描到问题的文件数
*  Author      : Kevin Zhang
*  History     :
*              1.create 2017-1-12
```



```
// 返回的格式如下：
*************************************************************************/
KEYSCANAPI_API int KS_ScanDir(const char *sInputDirPath,const char *sResultPath,int
nThreadCount=10,int bEncript=false);
```

4.10 KS_StatResultFilter

```
/*************************************************************************
*
*  Func Name  : KS_StatResultFilter
*
*  Description: 对扫描的统计结果进行过滤分析
*  Parameters : sFilename:输入的结果文件名
*                fThreshold：不良得分的阈值
*  Returns    : 成功扫描到问题的文件数
*  Author     : Kevin Zhang
*  History    :
*              1.create 2017-4-24
*************************************************************************/
KEYSCANAPI_API int KS_StatResultFilter(const char *sInputFilename, const char
*sResultFilename, float fThreshold = 5.0);
```

4.10 KS_GetLastErrorMsg

```
/*************************************************************************
 *
 *  Func Name  : KS_GetLastErrorMsg
 *
 *  Description: 返回最后出错的消息
 *                The function must be invoked while you needn't any lexical anlysis
 *
 *  Parameters : 返回最后出错的消息
 *
 *  Returns    : success or fail
 *  Author     : Kevin Zhang
 *  History    :
 *              1.create 2002-8-6
 *************************************************************************/
KEYSCANAPI_API const char* KS_GetLastErrorMsg();
```

4.11 KS_ExportDict 该接口不适用于一般开发接口


```
/*************************************************************************
*
*  Func Name  : KS_ExportDict
*
*  Description: ExportDict dictionary 导出已经定义的不良词词典
*                为保护知识产权，该功能仅局限于管理员内部调度使用
*  Parameters :
*                sFilename:Text filename for user dictionary
*                KS_HANDLE handle:  handle of KeyScanner
*  Returns    : The  number of  lexical entry imported successfully
*               成功导入的词典条数
*  Author     : Kevin Zhang
*  History    :
*              1.create 2014-8-3
*  文本文件的格式为:   词条    词类  权重 (注意，最多定义255个类别)
*        例如:  AV电影 色情 2
*                 六合彩 涉赌 8 1
*************************************************************************/

KEYSCANAPI_API int KS_ExportDict(const char *sFilename, KS_HANDLE handle = 0);
```

5. Java 调用的 JNA 接口

5.1 jna 使用说明

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 九眼智能过滤运行环境

6.1 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

6.2 Linux 如何调用九眼智能过滤

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/KeyScanAPI.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

7 作者简介

张华平 博士 副教授 研究生导师
大数据搜索挖掘实验室（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：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(Assitant)
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/
Webchat: drkevinzhang
Subscriptions: Thousands of Big Data Experts