

个人资料



buaalei

访问: 166921次
积分: 2314
等级: 5
排名: 第10483名
原创: 41篇 转载: 28篇
译文: 17篇 评论: 35条

文章搜索

文章分类

机器学习 (8)
Caffe (1)
C/C++ (31)
数据挖掘 (6)
Web (5)
Java (4)
技术趋势 (3)
OS/Unix/Linux (9)
Android (17)
NetWork (2)
视频编解码 (0)

文章存档

2016年03月 (1)
2015年08月 (1)
2015年07月 (2)
2015年06月 (5)
2015年05月 (3)

阅读排行

Weka -- 数据格式基本介 (9003)
Android 使用自定义cursor (7401)
Android ListPreference (7240)
谷歌大脑科学家 Caffe 的 (7186)
Weka -- 分类算法之C4.5 (5850)

[【免费公开课】模板匹配的字符识别\(OCR\)算法原理](#) [有奖试读—漫话程序员面试求职、升职加薪、创业与生活](#)

VC 下Microsoft Speech SDK开发小结

标签: [microsoft](#) [null](#) [xml](#) [stream](#) [input](#) [tts](#)

2010-03-12 09:04 5280人阅读 评论(0) 收藏 举报

分类: [C/C++ \(30\)](#)

1.首先开发得需要Microsoft Speech SDK的支持，以下是下载地址

<http://msdn.microsoft.com/code/sample.asp?url=/msdn-files/027/000/781/msdncompositedoc.xml>
Speech SDK 5.1 (68 MB)
<http://download.microsoft.com/download/speechSDK/SDK/5.1/WXP/EN-US/speechsdk51.exe>
5.1 Language Pack (81.5 MB)
<http://download.microsoft.com/download/speechSDK/SDK/5.1/WXP/EN-US/speechsdk51LangPack.exe>
Redistributables (128 MB)
<http://download.microsoft.com/download/speechSDK/SDK/5.1/WXP/EN-US/speechsdk51MSM.exe>
Documentation (2.28 MB)
<http://download.microsoft.com/download/speechSDK/SDK/5.1/WXP/EN-US/sapi.chm>

2.下载后，执行安装

下载完毕后首先安装SpeechSDK51.exe，然后安装中文语言补丁包SpeechSDK51LangPack，然后展开speechsdk51MSM.exe，这些都是自解压文件，解压后执行相应的setup程序到你要的目录，默认C:/Microsoft Speech SDK 5.1.对应的开发参考手册为sapi.chm，详细描述了各个函数的细节等。

3.VC的环境配置

在应用SDK的开发前当然得需要对工程环境进行配置，我用的是VS2003(其他情况类似)，配置的过程如下：

工具->选项->项目->VC++目录，在"显示以下内容的目录"下拉框中选择"包含目录"项，添加一项C:/Program Files/Microsoft Speech SDK 5.1/Include到目录中去。再选择"库文件"项，添加一项C:/Program Files/Microsoft Speech SDK 5.1/Lib/1386到目录中去。

4.其他准备项

基础的配置已经完成，那么接下来的工作就是要包含编译的头文件了，所以先将头文件和库文件包含进来

```
#include <sapi.h>
#include <sphelper.h>
#include <spuihelp.h>

#pragma comment(lib,"ole32.lib") //CoInitialize CoCreateInstance需要调用ole32.dll
#pragma comment(lib,"sapi.lib") //sapi.lib在SDK的lib目录,必需正确配置
```

具体其他函数所需要的头文件可参考sapi.chm手册。

5.源文件修改项

看上去上面的部分配置完成后就大功告成了，其实还不全是，当你编译时就会出错：

```
c:/program files/microsoft speech sdk 5.1/include/sphelper.h(769) : error C4430: missing type specifier - int assumed. Note: C++ does not support default-int
```

VC 下Microsoft Speech : (5279)
数据挖掘领域十大经典算 (5143)
关于线性空间及机器学习 (4936)
认知信息处理系统的认识 (4854)
Weka -- 聚类算法之K-me (4660)

评论排行

caffe在windows编译工程 (9)
Android 使用自定义curs (4)
Android--广播机制 (4)
谷歌大脑科学家 Caffe编 (3)
Weka -- 分类算法之C4.5 (2)
GetCurrentDirectory 和 (2)
menu.addIntentOptions ; (2)
C++动态绑定及继承 (2)
Weka --- 关联规则挖掘 (1)
Andorid 界面风格和主题 (1)

推荐文章

*4月28--30日：一起去北展揭开电信网络诈骗的神秘面纱
*Spring和MyBatis实现数据的读写分离
* Android几种常见的多渠道(批量)打包方式介绍
*Unity Native Render Plugin在VR中的绘制(二): 透明排序
*随机过程--Metropolis-Hastings 算法
* Android源码浅析 (一) ——VMware Workstation Pro和Ubuntu Kylin 16.04 LTS安装配置

最新评论

caffe在windows编译工程及运行渣码: 感谢博主的汇总, 资料很全。
caffe在windows编译工程及运行buaalei: @fd_318:看你这描述应该是数据有问题, leveldb格式数据, 要修改caffe训练的目录地址...
caffe在windows编译工程及运行buaalei: @mmmmmtttttf:运行的时候需要加命令行添加参数的, 调试->命令参数 添加train -s...
caffe在windows编译工程及运行buaalei: @qq_31696011:编译选项不对吧? 编译目标可以选x86或者x64
caffe在windows编译工程及运行buaalei: @qq_31696011:把依赖的环境配置好就可以了, 工程文件下载后就可以用了, 自己用的就是这个
谷歌大脑科学家 Caffe缔造者 贾quotong1988: 赞!!!
caffe在windows编译工程及运行fd_318: 博主。我在测试MNIST的时候出现【2600 data_layer.cpp:83】Unknown d...
谷歌大脑科学家 Caffe缔造者 贾freezenfire: 很不错, 获取很多有用的信息
谷歌大脑科学家 Caffe缔造者 贾andeyeluguo: 不错
Weka --- 关联规则挖掘

c:/program files/microsoft speech sdk 5.1/include/sphelper.h(1419) : error C4430: missing type specifier - int assumed. Note: C++ does not support default-int
c:/program files/microsoft speech sdk 5.1/include/sphelper.h(2373) : error C2065: 'psz' : undeclared identifier
c:/program files/microsoft speech sdk 5.1/include/sphelper.h(2559) : error C2440: 'initializing' : cannot convert from 'CSpDynamicString' to 'SPPHONEID *'
No user-defined-conversion operator available that can perform this conversion, or the operator cannot be called
c:/program files/microsoft speech sdk 5.1/include/sphelper.h(2633) : error C2664: 'wcslen' : cannot convert parameter 1 from 'SPPHONEID *' to 'const wchar_t *'
Types pointed to are unrelated; conversion requires reinterpret_cast, C-style cast or function-style cast
Speech代码编写时间太早, 语法不严密。而VS2003(及以上)对于语法检查非常严格, 导致编译无法通过。修改头文件中的以下行即可正常编译:

Line 769

修改前: const ulLenVendorPreferred = wcslen(pszVendorPreferred);
修改后: const unsigned long ulLenVendorPreferred = wcslen(pszVendorPreferred);

Line 1418

修改前: static CoMemCopyWFEX(const WAVEFORMATEX * pSrc, WAVEFORMATEX ** ppCoMemWFEX)
修改后: static HRESULT CoMemCopyWFEX(const WAVEFORMATEX * pSrc, WAVEFORMATEX ** ppCoMemWFEX)

Line 2372

修改前: for (const WCHAR * psz = (const WCHAR *)lParam; *psz; psz++) {}
修改后: const WCHAR * psz; for (psz = (const WCHAR *)lParam; *psz; psz++) {}

Line 2559

修改前: SPPHONEID* pphoneId = dsPhoneId;
修改后: SPPHONEID* pphoneId = (SPPHONEID*)((WCHAR *)dsPhoneId);

Line 2633

修改前: pphoneId += wcslen(pphoneId) + 1;
修改后: pphoneId += wcslen((const wchar_t *)pphoneId) + 1;

好了, 编译通过, 下面可以正式编写程序了。

6.SAPI实现TTS(Text to Speech)

- 1. 首先要初始化语音接口, 一般有两种方式:

```
ISpVoice* pVoice;  
::CoInitialize(NULL);  
HRESULT hr =CoCreateInstance(CLSID_SpVoice, NULL, CLSCTX_ALL,IID_ISpVoice, (void **)&pVoice);  
然后就可以使用这个指针调用SAPI函数了, 例如  
pVoice->SetVolume(50);//设置音量  
pVoice->Speak(str.AllocSysString(),SPF_ASYNC,NULL);
```

另外也可以使用如下方式:

```
CComPtr<ISpVoice> m_cpVoice;  
HRESULT hr = m_cpVoice.CoCreateInstance(CLSID_SpVoice );  
在下面的例子中都用了这个m_cpVoice变量。CLSID_SpVoice的定义位于sapi.h中。
```

- 2. 获取/设置输出频率。

SAPI朗读文字的时候, 可以采用多种频率方式输出声音, 比如: 8kHz 8Bit Mono、8kHz 8BitStereo、44kHz 16BitStereo等,在音调上有所差别。具体可以参考sapi.h。

可以使用如下代码获取当前的频率配置:

```
CComPtr<ISpStreamFormat> cpStream;  
HRESULT hrOutputStream =m_cpVoice->GetOutputStream(&cpStream);  
if (hrOutputStream ==S_OK)  
{  
    CSpStreamFormat Fmt;  
    hr = Fmt.AssignFormat(cpStream);  
    if (SUCCEEDED(hr))  
    {  
        SPSTREAMFORMAT eFmt = Fmt.ComputeFormatEnum();  
    }  
}
```

ording11: 请问楼主, 在进行关联规则时, 如果某个属性列有多个值, 而所挖掘的频繁项集就是里面的值, 这样的数据在w...

```
}
```

SPSTREAMFORMAT 是一个ENUM类型, 定义位于sapi.h中, 这样eFmt就保存了获得的当前频率设置值。每一个值对应了不同的频率设置。

通过如下代码设置当前朗读频率:

```
CComPtr<ISpAudio> m_cpOutAudio; //声音输出口
```

```
SpCreateDefaultObjectFromCategoryId( SPCAT_AUDIOOUT,&m_cpOutAudio ); //创建接口
```

SPSTREAMFORMAT eFmt = SPSF_8kHz8BitMono; //SPSF_8kHz 8Bit Mono这个参数可以参考sapi.chm手册

```
CSpStreamFormat Fmt;
```

```
Fmt.AssignFormat(eFmt);
```

```
if (m_cpOutAudio )
```

```
{
```

```
    hr = m_cpOutAudio->SetFormat(Fmt.FormatId(), Fmt.WaveFormatExPtr() );
```

```
}
```

```
else hr = E_FAIL;
```

```
if(SUCCEEDED( hr ) )
```

```
{
```

```
    m_cpVoice->SetOutput( m_cpOutAudio, FALSE );
```

```
}
```

- 3. 获取/设置播放所用语音。

引擎中所用的语音数据文件一般保存在SpeechEngines下的spd或者vce文件中。安装sdk后, 在注册表中保存了可用的语音, 比如英文的男/女, 简体中文的男音等。位置是:

HKEY_LOCAL_MACHINE/Software/Microsoft/Speech/Voices/Tokens

SAPI的缺点是不能支持中英文混读, 在朗读中文的时候, 遇到英文, 只能逐个字母读出。所以需要程序自己进行语音切换。

- (1) 可以采用如下的函数把当前SDK支持的语音填充在一个组合框中:

```
// SAPI5helper function in sphelper.h
```

```
CWnd* m_wnd = GetDlgItem(IDC_COMBO_VOICES);
```

```
HWND hWndCombo = m_wnd->m_hWnd; //组合框句柄
```

```
HRESULT hr = SpInitTokenComboBox( hWndCombo , SPCAT_VOICES );
```

这个函数是通过IEnumSpObjectTokens接口枚举当前可用的语音接口, 把接口的说明文字添加到组合框中, 并且把接口的指针作为LPARAM 保存在组合框中。

一定要记住最后程序退出的时候, 释放组合框中保存的接口:

```
SpDestroyTokenComboBox( hWndCombo );
```

这个函数的原理就是逐个取得combo里面每一项的LPARAM数据, 转换成IUnknown接口指针, 然后调用Release函数。

- (2) 当组合框选择变化的时候, 可以用下面的函数获取用户选择的语音:

```
ISpObjectToken* pToken = SpGetCurSelComboBoxToken( hWndCombo );
```

- (3) 用下面的函数获取当前正在使用的语音:

```
CComPtr<ISpObjectToken> pOldToken;
```

```
HRESULT hr = m_cpVoice->GetVoice( &pOldToken);
```

- (4) 当用户选择的语音和当前正在使用的不一致的时候, 用下面的函数修改:

```
if(pOldToken != pToken)
```

```
{
```

```
    // 首先结束当前的朗读, 这个不是必须的。
```

```
    HRESULT hr = m_cpVoice->Speak( NULL, SPF_PURGEBEFORESPEAK, 0);
```

```
    if (SUCCEEDED (hr) )
```

```
        hr = m_cpVoice->SetVoice( pToken );
```

```
}
```

- (5) 也可以使用函数SpGetTokenFromId获取指定voice的Token指针, 例如:

```
WCHAR pszTokenId[]
```

```
=L"HKEY_LOCAL_MACHINE//Software/Microsoft//Speech//Voices//Tokens//MSSimplifiedChineseVoice";
```

```
SpGetTokenFromId(pszTokenID , &pChineseToken);
```

- 4 开始/暂停/恢复/结束当前的朗读

要朗读的文字必须位于宽字符串中,所以从文本框中读取的字符串类型CString必须转换成为WCHAR型, 如下(m_strText为文本框变量):

```
CString strSpeak;  
m_strText.GetWindowText(strSpeak);  
WCHAR wChar[256];  
memset(wChar,0,256);  
MultiByteToWideChar( CP_ACP, 0, strSpeak, strSpeak.GetLength(), wChar, 256);
```

这样就将文本框中的字符串strSpeak转化为WCHAR型的wChar变量中了.

开始朗读的代码:

```
hr=m_cpVoice->Speak( wChar, SPF_ASYNC |SPF_IS_NOT_XML, 0 );
```

如果要解读一个XML文本, 用:

```
hr=m_cpVoice->Speak( wChar, SPF_ASYNC |SPF_IS_XML, 0 );
```

暂停的代码: m_cpVoice->Pause();

恢复的代码: m_cpVoice->Resume();

结束的代码: (上面的例子中已经给出了)

```
hr=m_cpVoice->Speak( NULL, SPF_PURGEBEFORESPEAK,0);
```

- 5 跳过部分朗读的文字

在朗读的过程中, 可以跳过部分文字继续后面的朗读, 代码如下:

```
ULONG ulGarbage = 0;  
WCHAR szGarbage[] =L"Sentence";  
hr =m_cpVoice->Skip( szGarbage, SkipNum,&ulGarbage );
```

SkipNum是设置要跳过的句子数量, 值可以是正/负。

根据sdk的说明, 目前SAPI仅仅支持SENTENCE这个类型。SAPI是通过标点符号来区分句子的。

- 6 播放WAV文件。SAPI可以播放WAV文件, 这是通过ISpStream接口实现的:

```
CComPtr<ISpStream> cpWavStream;  
WCHAR szwWavFileName[NORM_SIZE] = L"";  
  
USES_CONVERSION;  
wcscpy( szwWavFileName, T2W(szFileName) );//从ANSI将WAV文件的名字转换成宽字符串  
  
//使用sphelper.h 提供的这个函数打开wav 文件, 并得到一个 IStream 指针  
hr = SPBindToFile(szwWavFileName, SPFM_OPEN_READONLY, &cpWavStream);  
if( SUCCEEDED( hr ) )  
{  
    m_cpVoice->SpeakStream( cpWavStream, SPF_ASYNC, NULL);//播放WAV文件  
}
```

- 7 将朗读的结果保存到wav文件

```
TCHARszFileName[256];//假设这里面保存着目标文件的路径  
USES_CONVERSION;  
WCHAR m_szWFileName[MAX_FILE_PATH];  
wcscpy( m_szWFileName,T2W(szFileName) );//转换成宽字符串
```

//创建一个输出流, 绑定到wav文件

```
CSpStreamFormat OriginalFmt;  
CComPtr<ISpStream> cpWavStream;  
CComPtr<ISpStreamFormat> cpOldStream;  
HRESULT hr =m_cpVoice->GetOutputStream(&cpOldStream );  
if (hr == S_OK) hr =OriginalFmt.AssignFormat(cpOldStream);  
else hr =E_FAIL;  
// 使用sphelper.h中提供的函数创建 wav文件  
if (SUCCEEDED(hr))  
{  
    hr = SPBindToFile( m_szWFileName, SPFM_CREATE_ALWAYS,&cpWavStream,&OriginalFmt.FormatId
```

```
( ), OriginalFmt.WaveFormatExPtr() );
}
if( SUCCEEDED( hr ) )
{
    //设置声音的输出到 wav 文件，而不是speakers
    m_cpVoice->SetOutput(cpWavStream, TRUE);
}
//开始朗读
m_cpVoice->Speak( wChar, SPF_ASYNC | SPF_IS_NOT_XML, 0 );

//等待朗读结束
m_cpVoice->WaitUntilDone( INFINITE );
cpWavStream.Release();

//把输出重新定位到原来的流
m_cpVoice->SetOutput( cpOldStream, FALSE );
```

- 8 设置朗读音量和速度

```
m_cpVoice->SetVolume((USHORT)hpos); //设置音量，范围是 0 -100
m_cpVoice->SetRate(hpos); //设置速度，范围是 -10 - 10
```

- 9 设置SAPI通知消息。

SAPI在朗读的过程中，会给指定窗口发送消息，窗口收到消息后，可以主动获取SAPI的事件，根据事件的不同，用户可以得到当前SAPI的一些信息，比如正在朗读的单词的位置，当前的朗读口型值（用于显示动画口型，中文语音的情况下并不提供这个事件）等等。要获取SAPI的通知，首先要注册一个消息：

```
m_cpVoice->SetNotifyWindowMessage( hWnd, WM_TTSAPPCUSTOMEVENT, 0, 0 );
```

这个代码一般是在主窗口初始化的时候调用，hWnd是主窗口（或者接收消息的窗口）句柄。

WM_TTSAPPCUSTOMEVENT是用户自定义消息。在窗口响应WM_TTSAPPCUSTOMEVENT消息的函数中，通过如下代码获取sapi的通知事件：

```
CSpEvent event; // 使用这个类，比用 SPEVENT结构更方便
```

```
while(event.GetFrom(m_cpVoice) == S_OK )
{
    switch( event.eEventId )
    {
        ...
    }
}
```

eEventID有很多种，比如SPEI_START_INPUT_STREAM表示开始朗读，SPEI_END_INPUT_STREAM表示朗读结束等。

可以根据需要进行判断使用。

7.总结

还有一些关于xml的支持可以参考sapi.chm帮助手册，感谢网络原作提供的资源，有iwaswzq, yaooo等。

顶 踩
1 0

上一篇 [UNIX进程概念小结\(apue\)](#)

下一篇 [XML介绍](#)

我的同类文章

C/C++ (30)

- mysql-connector-c++ +vs20... 2016-03-03 阅读 113
- VisualStudio 路径和项目属... 2015-06-03 阅读 1112
- CreateInstance(__uuidof(Co... 2014-05-17 阅读 2154
- windows下开发ffmpeg编译... 2012-02-27 阅读 2514
- 硬链接和符号链接简介 2012-02-14 阅读 2274
- Protocol Buffer技术详解(语... 2015-07-15 阅读 204
- LibUSB-Win32 在Windows7... 2015-05-14 阅读 3052
- VS2010下编译安装DarwinS... 2013-10-24 阅读 1221
- linux下添加PATH的方法 2012-02-15 阅读 2191
- C++操作符重载 2011-12-06 阅读 2127

更多文章

猜你在找

iOS8-Swift开发教程

VC++游戏开发基础系列从入门到精通

Qt基础与Qt on Android入门

MOS认证考试——PPT2013（考试代码77-422）

软件测试基础

Microsoft Speech Application SDK 版本 10 Beta 3

关于Microsoft Speech SDK 中TTS的研究

Hello World by Microsoft Speech SDK 51

一个简单的 Microsoft Speech SDK 51 例子

Hello World by Microsoft Speech SDK 51



查看评论

暂无评论

发表评论

用户名: MichaelLiang12

评论内容:

评论框

提交

* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

核心技术类目

全部主题 Hadoop AWS 移动游戏 Java Android iOS Swift 智能硬件 Docker OpenStack
VPN Spark ERP IE10 Eclipse CRM JavaScript 数据库 Ubuntu NFC WAP jQuery
BI HTML5 Spring Apache .NET API HTML SDK IIS Fedora XML LBS Unity
Splashtop UML components Windows Mobile Rails QEMU KDE Cassandra CloudStack FTC
coremail OPhone CouchBase 云计算 iOS6 Rackspace Web App SpringSide Maemo
Compuware 大数据 aptech Perl Tornado Ruby Hibernate ThinkPHP HBase Pure Solr
Angular Cloud Foundry Redis Scala Django Bootstrap

公司简介 | 招贤纳士 | 广告服务 | 银行汇款帐号 | 联系方式 | 版权声明 | 法律顾问 | 问题报告 | 合作伙伴 | 论坛反馈

网站客服 杂志客服 微博客服 webmaster@csdn.net 400-600-2320 | 北京创新乐知信息技术有限公司 版权所有 | 江苏乐知网络技术有限公司 提供商务支持

京 ICP 证 09002463 号 | Copyright © 1999-2014, CSDN.NET, All Rights Reserved

