

# 基于.net和SAPI的图片语音识别系统

曾 谦 四川大学计算机(软件)学院, 四川 成都 610064

**摘要:** 语音识别是计算机多媒体研究的一个热点。本文对语音识别技术进行了初步研究, 成功利用.net平台和借助微软SAPI接口实现了一个图片语音查找系统, 并给出了该系统实例。

**关键词:** SAPI;C#;.NET;语音识别

**中图分类号:** TP312 **文献标识码:** A **文章编号:** 1003-9767 (2010) 07-0115-02

## 1. 引言

语音识别是信息处理领域的一个重要方向, 语音识别通常又可以分为两种工作方式: 一种是连续语音听写, 这种方式需要语音识别引擎对语音听写过程中上下文以及相同相似发音的词语进行分析、做出判断, 实现难度较大。另一种是命令与控制方式(command and control), 在此种方式下, 语音识别引擎可以通过对简短的命令语音进行识别, 以便执行相应操作。

语音识别技术基础是非常复杂的。正是如此, 它获得了世界上著名的公司与研究机构的青睐, 语音识别技术正经历着日新月异的改变, 其中一些产品已经取得较好的效果, 用户可以在上述产品基础上进行二次开发。

## 2. 开发图片语音识别系统的条件

语音识别技术的发展涉及人机交互, 自然语言处理, 人工智能。这使开发人员难以从底层进行应用开发, 欣慰的是一些公司与研究机构已经推出他们的研究成果, 并且提供了供开发人员使用的开发平台与接口。其中影响比较大的有微软的SAPI、剑桥大学发布的HTK, IBM的via voice开发平台。本文采用微软的Speech SDK5.1开发包, 它不仅是一个免费开发包, 同时提供对中文的支持。

### 2.1 微软SAPI简介

微软的 Speech SDK是基于COM的视窗操作系统开发工具包。这个 SDK中含有语音应用程序接口(SAPI)、微软连续语音识别引擎(MCSR)以及串联语音合成(又称文本到语音)引擎(TTS)等等。SAPI 中还包括对于底层控制和高度适应性的直接语音管理、训练向导、事件、语法、编译、资源、语音识别管理以及文本到语音管理, 其结构如图1所示。

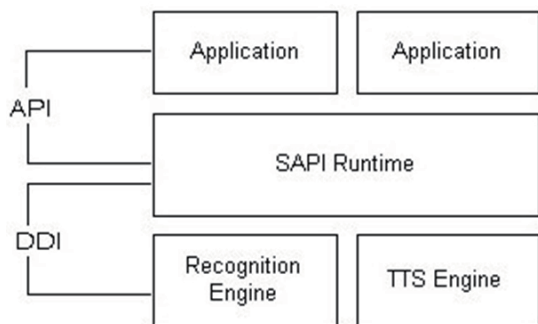


图1 SAPI结构

SAPI API在应用程序和语音引擎之间提供一个高级别的接口。SAPI 实现了所有必需的对各种语音引擎的实时的控制和管理等低级别的细节。

SAPI引擎的两个基本类型是文本语音转换系统(TTS)和语音识别系统。TTS系统使用合成语音合成文本字符串和文件到声音音频流。语音识别技术转换人类的声音语音流到可读的文本字符串或者文件。

### 2.2 在.net应用程序中使用SAPI的准备

安装SAPI 5.1语音开发包。

由于SAPI基于Windows平台, 通过COM接口供.net平台调用, 所以具体调用方法为: 新建工程, 在此工程的解决方案中单击右键, 在添加/引用中选择 Microsoft Speech Object Library, 并在程序中添加对SAPI命名空间的引用: using SpeechLib。

## 3. 系统模型设计及实现

### 3.1 创立系统界面

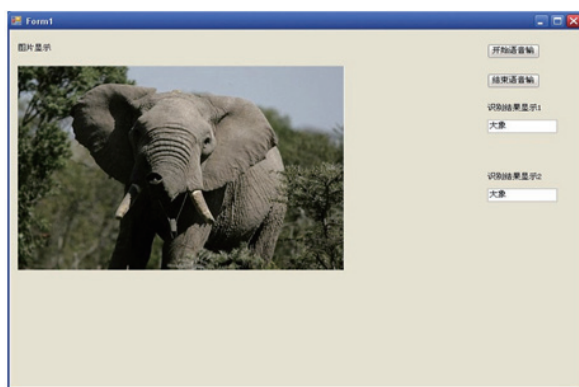


图2 系统界面一览

### 3.2 使用SAPI语音识别核心类

ISpRecognizer类是支持语音识别的核心类, 主要用于访问MCSR实现语音识别。在进行语音识别时, 微软Speech SDK 5.1 支

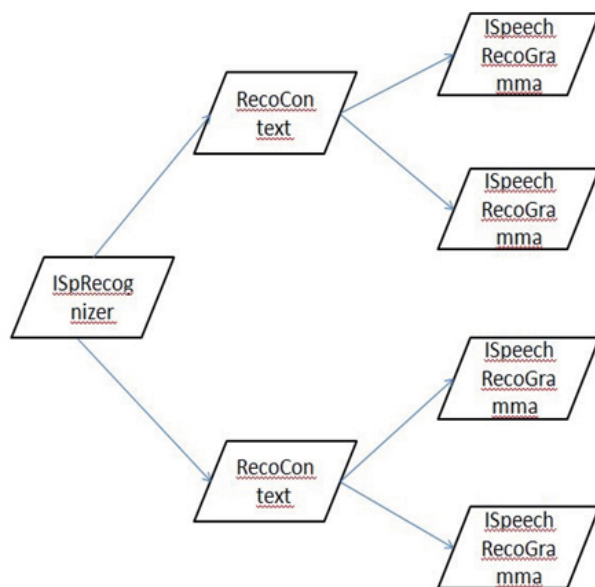


图3 语音识别类对应关系

持两种模式的语音识别引擎：共享（Share）和独享（InProc）。在本系统中，我们使用共享型语音识别引擎，CLSID\_SpSharedRecognizer。

SpSharedRecoContext 类是语音识别上下文接口，它可以发送与接收消息通知，通过CreateGrammar方法创建语法规则。通过对它的两个事件：\_ISpeechRecoContextEvents\_RecognitionEventHandler(ssrc\_Recognition)、\_ISpeechRecoContextEvents\_HypothesisEventHandler(Hypo\_Event)的重写，我们可以很轻松的得到当前语音识别结果。第一个事件对应的就是识别结果显示：1、第二个事件对应的就是识别结果显示；2、ISpeechRecoGrammar 类是语法规则接口，定义语音识别引擎需要是别的语音类容。起语法规则可以分为听写型与命令控制型。听写型可以识别大词汇量语音，但是别效率与识别准确率较低。命令控制型有针对性的对特定语音进行识别，在效率与准确率上都非常高。

当然一个ISpRecognizer不管是共享还是独享，都可以有多个RecoContext 与其关联，而一个RecoContext 也可以与多个ISpeechRecoGrammar 关联，每个ISpeechRecoGrammar 识别各自规定的内容。

### 3.3 编写系统核心代码

通过读入外部Grammar配置语音识别引擎，外部Grammar文件以xml文件格式存储。具体形式如下：

```
ManegePic.xml
<?xml version="1.0" encoding="gb2312" ?>
<GRAMMAR LANGID="804">
<RULE NAME="命令" TOPLEVEL="ACTIVE">
<L>
<P>玫瑰</P>
<P>大象</P>
<P>狮子</P>
<p>老虎</p>
<P>仙人掌</P>
<P>珠穆朗玛峰</P>
<P>布达拉宫</P>
<P>貂蝉</P>
<P>十字军</P>
<P>世界杯</P>
<P>阿里巴巴</P>
<P>乒乓球</P>
<P>五星红旗</P>
.....
</L>
</RULE>
</GRAMMAR>
```

采用命令控制型Grammar是为了使语音识别结果准确有效，定义外部Grammar是为了保持以后系统的可扩展性。对于新输入的图片，只要将图片名字加入到<P>图片名字</P>中即可。

开始语音输入事件：

```
private void btnStart_Click(object sender, EventArgs e)
{
    ssrc = new SpSharedRecoContext();
    srg = ssrc.CreateGrammar(1);
    srg.CmdLoadFromFile("ManegePic.xml", SpeechLoadOption.
    SLODynamic);
    //读入规则
```

```
    ssrc.Recognition += new _Ispeech hRecoC ontextEvents_Recognition
    ventHandler(ssrc_Recognition);
    //添加识别最终事件
    ssrc.Hypothesis += new _ISpeechRecoCon textEvents_HypothesisEve
    ntHandler(Hypo_Event);
    //添加识别怀疑事件
    srg.CmdSetRuleState(srg.Rules.Item(0).Name, SpeechRuleState.
    SGDSActive);
    //激活规则
}
```

其中识别最终事件是系统在完整处理输入音频之后得到的最后识别结果，怀疑事件则是因为系统在处理过程中的最相似识别结果。这两个事件在一些环境之下得到的识别结果不同。

识别怀疑事件：

```
private void Hypo_Event(int StreamNumber, object StreamPosition,
ISpeechRecoResult Result)
{
    textBox2.Text = Result.PhraseInfo.GetText(0, -1, true);
}
将系统在处理过程中的最相似结果显示在textBox控件之中。
识别最终事件：
void ssrc_Recognition(int StreamNumber, object StreamPosition,
SpeechRecognitionT ype RecognitionType, ISpeechRecoResult Result)
{
    textBox1.Text = Result.PhraseInfo.GetT ext(0, -1, true);
    // 将系统最终结果显示在textBox控件之中。
    try
    {
        Picturebox1.image = Result.PhraseInf o.GetText(0, -1, true).
        toString()+" .jpg ";
        //显示识别结果的图片
    }
    Catch (Exception e)
    {}
}
```

由于篇幅有限，这里仅仅给出了构建语音识别引擎与图片显示部分的代码。完整的系统可以在Windows XP + vs2008环境中调试通过，实现基于.net和SAPI的图片语音识别系统。

## 4. 结语

通过制作图片语音识别系统，对利用SAPI进行语音识别有了初步的研究和实践，该系统具有一定的扩展性，也可以作为子系统嵌入到其他系统之中。该系统的功能可以扩展到语音控制文件操作，比如打开/关闭文件，添加文件，可以制作一个不需要键盘鼠标只利用语音就可进行操作的软件系统。

## 参考文献：

- [1] Microsoft Speech 5.1 SDK Help .Microsoft Corporation
- [2] 童强 用Delphi实现基于SAPI的电脑语音控制应用程序[J].微计算机信息（测控自动化）.2006,22(3-1).
- [3] 宋阳.利用SAPI5 完成中文语音因素分解[J].计算机信息.2005,3:230-231