

文章编号: 1005 - 0523(2010)01 - 0052 - 06

基于 Web 的英语自助学习系统设计

张红斌¹ 李广丽²

(华东交通大学 1. 软件学院; 2. 信息工程学院, 江西 南昌 330013)

摘要: 分析了基于 Web 的英语自助学习系统的工作原理, 选择 Microsoft Speech SDK5.1 并在其基础之上封装 WebVoiceCtl 作为构建系统的核心技术。调用 Text To Speech Engine 完成文本语音转换, 实现基于 Web 的英语听力测试。此外, 通过 WebVoiceCtl 控件调用 Speech Recognition Engine 完成语音识别, 实现基于 Web 的英语口语测试。最后, 在 ASP.NET 环境下实现英语自助学习软件系统, 该软件可提供基于 Web 的英语听、说、写等技能训练。

关键词: 英语自助学习系统; 文本语音转换; 语音识别; Microsoft Speech SDK; WebVoiceCtl

中图分类号: TP391

文献标识码: A

计算机辅助学习已经成为时下流行的学习模式, 英语的学习也不例外。例如, 人们在学习英语或阅读英语文章时常会遇到生词, 很多人会立即求助于词库软件进行单词查询。目前流行的词库软件多为单机版本, 如谷歌金山词霸、灵格斯翻译等, 它们的使用具有一定的局限性, 即无法面向 Web 进行单词查询、语音表达、口语纠正以及学习资源共享等。故本文叙述了在 Web 环境下如何构造 1 个英语自助学习软件系统。

英语学习强调“听”“说”“写”3 个方面的技能训练^[1], 中国学生的英语阅读能力普遍较好, 但由于缺乏语言环境, 他们的听力和口语能力却有待提高。因此, 如何利用 Web 网站技术、多媒体技术、语音识别技术完成 1 个可以面向 Web 用户的开放式英语学习系统, 并侧重于听力及口语训练, 对于提高中国学生的英语综合素质是 1 件十分有意义的事情。

1 基于 Web 的英语自助学习系统的工作原理

英语自助学习系统的核心功能是听力训练和口语训练, 实现的关键技术是文本语音转换技术(Text To Speech, TTS) 和语音识别技术(Speech Recognition, SR)。文语转移 TTS 的主要功能是将计算机中任意出现的文字转换成自然流畅的语音输出^[2]。运用 TTS 技术可实现英文单词、词组、句子及文章的在线朗读, 指引学生完成基于 Web 的英语听力训练。运用 SR 技术则可实现学生所朗读单词、句子的实时语音识别, 指引学生完成基于 Web 的英语口语训练。目前, 国内著名网站 nciku 已经实现了基于 Web 的英文文本语音转换, 但却无法完成英语口语的实时识别及诊断。

微软开发了 1 套基于 Windows 环境的语音编程专用组件 Microsoft Speech SDK5.1, 该组件包含了语音识别(Speech Recognition) 和语音合成(Speech Synthesis, SS) 两大技术^[1]。它可以为我们提供清晰和标准的美式英语发音和标准普通话的简体中文男生发音^[3]。该组件包是实现基于 Web 的英语自助学习系统的重要基础。

1.1 文本语音转换的工作原理

在 Speech SDK 中 TTS 功能的工作原理如图 1 所示。语音应用程序编程接口 SAPI 是介于语音应用程序和语音引擎之间的中间层^[4]。其中 SpVoice 对象需要实现 ISpVoice 和 ISpTTS EngineSite 两个接口^[2]。ISpVoice 是应用程序能正常操作 TTS 功能的接口, 它是控制文本语音转换的关键。ISpTTS EngineSite 用来

收稿日期: 2009 - 08 - 28

基金项目: 江西省教育厅科学技术项目(2006 [177]); 华东交通大学科学技术研究项目(08XX05)

作者简介: 张红斌(1979 -), 男, 硕士, 讲师, 研究方向为图像处理、多媒体技术等。

写音频数据和消息队列事件。TTS Engine 实现 ISpTTSEngine 和 ISpObjectWithToken 两个接口。ISpTTSEngine 是 SAPI 调用 TTS Engine 的接口^[5],而 ISpObjectWithToken 使得 SAPI 可以创建和初始化 TTS Engine。如果用户软件存在界面(User Interface)则必须实现 ISpTokenUI 接口^[1]。SAPI 在应用程序和语音引擎之间提供了 1 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控制和管理等低级别细节。SAPI 还支持对 SpVoiceObject 以 ActiveX 控件的形式进行调用,故在基于 Web 的英语自主学习系统中,当网页加载时,同时加载 SpVoiceObject 就可完成基于 Web 的文本语音转换功能。

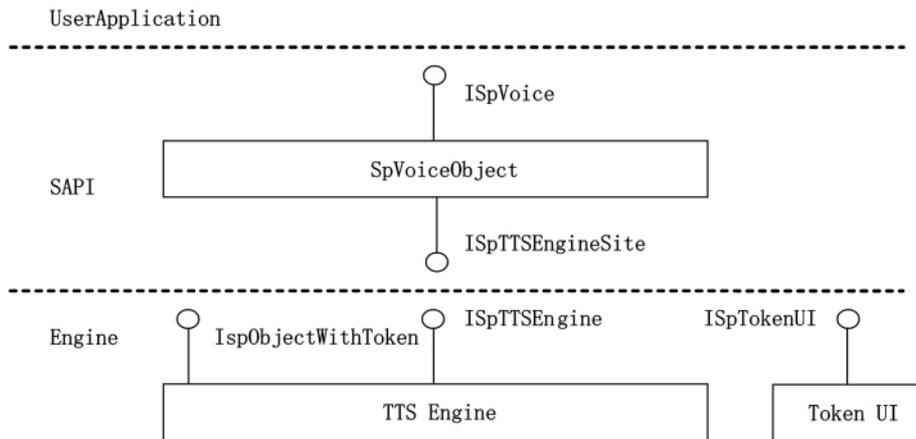


图 1 在 Speech SDK 中的 TTS 原理

1.2 语音识别的工作原理

ISpRecoContext 是语音识别的主接口,它是语音程序接收被请求的语音识别事件通知的媒介^[5]。创建 1 个共享的 ISpRecoContext 接口后,即可完成语音识别的环境设置,而要创建该接口则必须先创建 1 个 SAPI 默认的音频输入流,然后设置语音识别事件通知,其中最重要的事件通知是 SPEI - RECOGNITION。最后,创建应用程序,加载并激活 1 个 ISpRecoGrammar,即指定被识别的语音规则。

要实现基于 Web 的语音识别,其转换引擎必须以 ActiveX 的方式嵌入到网页中^[6]。加拿大的 Geoff Bailey 基于 Speech SDK 5.1 和 ATL 技术封装了 1 个可完成语音识别的 ActiveX 控件 WebVoiceCtl,该控件可以嵌入到网页中,接受用户的语音输入,然后根据 Grammar 完成对指定单词、句子的识别,即识别满足正确发音标准的英语语音^[7]。

WebVoiceCtl 的语音识别工作流程如图 2 所示。待识别的内容存储在 1 个 XML 文件 WebVoice.xml 中,程序人员可以对其进行编程把需要识别的单词、句子等装载进去,为语音识别做好准备。识别的语音规则被存储在 Grammar.xml 中,当 WebVoiceCtl 控件被成功编译后,该规则即被装载到控件中,不能再修改,故必须在编译生成控件前就设定好语音规则。在语音识别过程中,语音识别引擎会把输入的用户语音与正在运行的语音规则进行比较,如果识别规则匹配,则提示正确识别,否则提示用户重新输入语音信息。



图 2 WebVoiceCtl 控件语音识别功能的流程图

2 基于 Web 的英语自主学习系统的设计

基于 Web 的英语自主学习系统综合运用了 TTS、SR、XML 和数据库等技术,其中关键技术的集成如图 3 所示。软件在使用前客户端必须首先正常安装 Microsoft Speech SDK 5.1。当 TTSPage. AspX 网页加载到客户端浏览器后,即通过 JavaScript 脚本实现 1 个 Spvoice 对象,该对象内嵌在网页中,当用户检索英语词库获取单词、词组、句子等英文文本信息时,Spvoice 对象就会调用客户端电脑的 TTS 引擎,执行 Speak 方法将文本信息转换为语音信息,并通过客户端电脑的扬声器发出。

在 SRPage. AspX 网页中内嵌了 WebVoiceCtl 控件,它必须首先在客户端的注册表中完成注册。当客户端与服务器连接后,服务器立即为该客户端复制 1 份 WebVoice. xml 文件,用户可以输入英文单词、词组、句子等,并将其注入到 WebVoice. xml 文件中,在语音识别引擎初始化后,WebVoiceCtl 接受客户端语音输入,然后读取 WebVoice. xml 文件,遵循图 2 的流程完成语音识别功能。

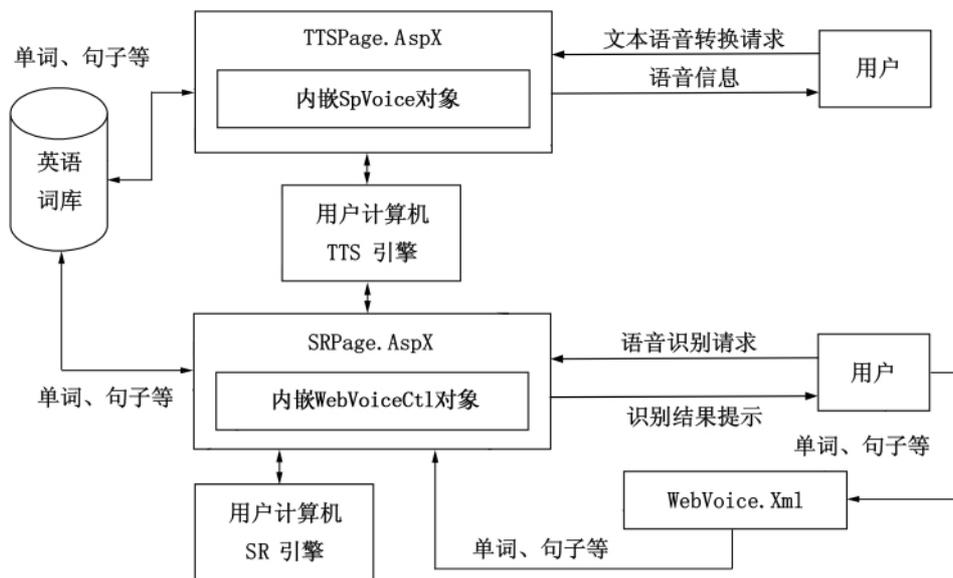


图 3 基于 Web 的英语自主学习系统的关键技术集成示意图

3 基于 Web 的英语自主学习系统的实现

软件开发环境: Microsoft Visual Studio 2005 + SQL Server 2000 + Microsoft Speech SDK 5.1 + WebVoiceCtl + IIS + XML, 在 Win2000 SP4/Win XP SP2 下均可正常运行。

3.1 文本语音转换功能的实现

文本语音转换功能的软件实现思想如下(仅以单词的 TTS 为例,词组、句子等的 TTS 处理方法类似)。

第 1 步 通过脚本创建 SpVoice 对象,并将其加载到用户浏览器中,脚本代码: `var VoiceObj = new ActiveXObject("SAPI.SpVoice")`。

第 2 步 创建触发 TTS 功能的图片按钮: ``。

第 3 步 实现 SpeakText 方法,将文本内容转换为语音: `VoiceObj. Speak($(TxtWord). value, 1)`。

在英文文本朗读的同时,用户还可设置朗读者的朗读口型。微软设计了 1 组较合理的口型图片顺序^[1],该口型图片序列能较准确地模拟人的英语发音口型,脚本代码如下。其中 VisemeId 为口型编号,它从 1 变化到 21,循环反复。

```
function VoiceObj. : Viseme( StreamNum, StreamPos, Duration, VisemeType, Feature, VisemeId ) {
```

```

// Map the VisemeId to the appropriate. bmp
if( VisemeId == 15 || VisemeId == 17 || VisemeId == 18 || VisemeId == 21 )

idImage. src = "picture/mouthop1. bmp"; //对应嘴型 

else if( VisemeId == 14 || VisemeId == 16 || VisemeId == 19 || VisemeId == 20 )

idImage. src = "picture/mouthop2. bmp"; //对应嘴型 

else if( VisemeId == 4 || VisemeId == 6 || VisemeId == 9 || VisemeId == 12 )

idImage. src = "picture/mouthop3. bmp"; //对应嘴型 

else if( VisemeId == 1 || VisemeId == 2 || VisemeId == 3 || VisemeId == 11 )

idImage. src = "picture/mouthop4. bmp"; //对应嘴型 

else if( VisemeId == 7 || VisemeId == 8 )

idImage. src = "picture/mouthnar. bmp"; //对应嘴型 

else if( VisemeId == 5 || VisemeId == 10 || VisemeId == 13 )

idImage. src = "picture/mouthmed. bmp"; //对应嘴型 

else

idImage. src = "picture/mouthclo. bmp"; //对应嘴型 

}

```

此外,用户可基于 Web 进行转换语音的语速、音量、朗读者等属性设置,以优化其人机交互界面,脚本代码不再叙述,可见 Microsoft Speech SDK 的帮助文档。

3.2 语音识别功能的实现

以单个英语单词的语音识别为例,词组、句子及文章的识别依赖于逐个英文单词的正确识别,故英文单词的语音识别是整个语音识别功能的关键。

第 1 步 创建 SRPage. AspX 并将 WebVoiceCtl 控件嵌入到网页中。当客户端发出语音识别请求后,它可以响应并正确处理,其代码如下

```

<object id = "voice" classid = "CLSID: 143D27F4 - F40B - 4F28 - 918B - FC156A72C07D" codebase = "http://www3. te-
lus. net/CanGeordie/bin/WebVoiceCtl. dll" type = "application/x - oleobject" > </object >

```

第 2 步 初始化语音识别转换引擎。其代码被封装在 WebVoiceCtl 控件中,是 C + + 实现,Geoff Bailey 已经将其封装好,以下均为引用。语音识别引擎通过按钮点击触发初始化函数,代码如下

```

LRESULT CWebVoice:: OnBtnVoice( WORD wNotifyCode ,WORD wID ,HWND hWndCtl ,BOOL& bHandled)
{
    bVoiceBtn = ! bVoiceBtn;
    InvalidateRect( NULL);
    if( bVoiceBtn) {
        if( ! bInit) { //如果当前还未进行初始化操作
            :: UpdateWindow( hWndCtl);
            if( FAILED( InitSapi( ))) { bVoiceBtn = ! bVoiceBtn; Invalidate( ); return 1; //未能正常初始化,则立刻返回}
        }
        else m - cpRecoCtxt - > Resume( NULL); //若语音识别引擎已初始化,则恢复语音识别
        SetDlgItemText( IDC - TXT - VOICE ( unsigned short * ) " Ready to navigate: "); //设置按钮
    }
}

```

```

//文字
}
else {
    if( bInit) m - cpRecoCtxt - > Pause( NULL ); //暂停
    SetDlgItemText( IDC - TXT - VOICE ( unsigned short * ) " Paused" ); //设置按钮文字
}
bHandled = true;
return 0;
}

```

第3步 注入识别单词。用户选择将待识别的单词注入到 WebVoice.xml 中,为语音识别做好准备,这是提供给二次开发者的编程接口,实现代码如下

```

protected void Button1 - Click( object sender ,EventArgs e)
{
    string inputWord = txtInputWord.Text; //从文本框获取用户输入的单词
    XmlDocument xmlDoc = new XmlDocument(); //创建 1 个 xml 文档对象
    xmlDoc.Load( Server.MapPath( " data/WebVoice.xml" ) ); //载入待注入单词的文件
    XmlNodeList nodelist = xmlDoc.GetElementsByTagName( " item" ); //获取 item 结点
    int i = 0;
    foreach( XmlNode xn in nodelist ) //循环遍历 item 结点列表
    {
        XmlElement xe = ( XmlElement ) xn; //将其强制转换为 XmlElement 类型
        XmlNodeList nls = xe.ChildNodes; //获取当前 item 结点的子结点
        if ( i == 3 ) //注入单词的起始地址
        {
            foreach ( XmlNode xn1 in nls )
            {
                XmlElement xe2 = ( XmlElement ) xn1;
                if ( xe2.Name == " txt" ) //在 item 的第 1 个子 txt 结点中注入单词内容
                {
                    xe2.InnerText = inputWord;
                    break; //退出当前循环
                }
            }
            i++; //循环遍历结点
        }
        xmlDoc.Save( Server.MapPath( " data/WebVoice.xml" ) ); //保存修改后的 Xml 文档
    }
}

```

第4步 完成语音识别。WebVoiceCtl 虽然内嵌在网页中,但它却是 1 个 ActiveX 控件,可以响应 Windows 消息,故在其封装代码中,首先为语音识别指定消息响应列表 MESSAGE - HANDLER(WM - RECOEVENT ,OnRecoEvent) ,即当该控件接受到 WM - RECOEVENT 消息后,立即执行 OnRecoEvent 函数,而在 OnRecoEvent 函数中通过 ProcessRecoEvent 函数处理语音识别,ProcessRecoEvent 函数代码如下,以下均为引用。

```

void CWebVoice:: ProcessRecoEvent( HWND hWnd )
{
    CSpEvent event; //定义识别事件
    while ( event.GetFrom( m - cpRecoCtxt ) == S - OK ) {

```

```

switch ( event. eEventId) {
case SPEI - RECOGNITION: //正确的识别
    ExecuteCommand( event. RecoResult( ) ,hWnd) ; //正确识别时的处理
    break; //结束识别处理
case SPEI - FALSE - RECOGNITION: //错误的识别
    HandleFalseReco( event. RecoResult( ) ,hWnd) ; //错误识别时的处理
    break; //结束识别处理
default:
    break; //结束识别处理
}}

```

4 结论

基于 Web 的英语自主学习系统使学习者足不出户就可轻松完成英语学习,其学习内容涵盖了英语学习的听、说、写等训练,并强调了听力训练和口语训练的重要性,该系统具有较好的通用性和推广价值。日后,笔者仍会在语音识别的效率以及多语种自助学习的 Web 应用等方面做进一步的研究,期望有更多的智能化的语言学习系统能够得到应用,以满足更加丰富、多样的语言学习需求。

参考文献:

- [1] 李广丽,张红斌. 基于 TTS 和 SR 技术的英语自学系统的设计[J]. 华东交通大学学报, 2009, 26(2): 86-90.
- [2] 廖日坤,纪越峰,黄小迅. 基于 TTS 文语转换的 Web 语音浏览器[J]. 兰州工业高等专科学校学报, 2006, 13(2): 10-13.
- [3] 赵强,左娅佳,房维强,等. 诊断学仿真实验系统中应用 TTS 技术创建虚拟问诊功能的研发设计[J]. 电化教育研究, 2009, (10): 62-63.
- [4] 林茜,欧建林,蔡骏. 基于 Microsoft Speech SDK 的语音关键词检出系统的设计与实现[J]. 心智与计算, 2007, 1(4): 433-441.
- [5] 袁军,张思民. 嵌入式中文 TTS 系统的研究与实现[J]. 电脑知识与技术, 2008, (6): 1345-1347.
- [3] 尹惠玲,杨帆,于虹,等. 基于 COM 的智能 TTS 系统的设计与实现[J]. 微计算机信息, 2009, 25(5-3): 172-173.
- [4] GEOFF B. Voice - activated Web Browsing[EB/OL]. 2004 [2009-08-16]. <http://www.codeproject.com/KB/audio-video/WebVoicePkg.aspx>.

A Design of an English Self - study System Based on Web

Zhang Hongbin¹, Li Guangli²

(1. School of Software; 2. Information Engineering School, East China Jiaotong University, Nanchang 330013 China)

Abstract: The paper analyzes the working theory of an English self - study system based on web. Microsoft Speech SDK 5.1 and an Activex control named WebVoiceCtl which is encapsulated by Geoff Bailey with the help of the Speech SDK 5.1 are chosen to build the English self - study system. Text To Speech Engine is chosen to complete the text to speech function which is the key technology of the English listening testing on web. Moreover, the WebVoiceCtl is chosen to complete the speech recognition function which is the key technology of the English oral testing on web. Finally, the English Self - Study System based on web is designed successfully in the environment of ASP.NET which offers trainings of English listening, speaking, reading and writing.

Key words: English self - study system; text to speech; speech recognition; Microsoft Speech SDK; WebVoiceCtl

(责任编辑 刘棉玲)