基于 HTK 的特定词语音识别系统①

曾 妮，费洪晓，姜振飞

(中南大学 软件学院，长沙 410075)

摘 要：语音识别技术经过半个世纪的发展，目前已日趋成熟，其在语音拨号系统、数字遥控、工业控制等领域都有了广泛的应用。由于目前常用的声学模型和语言模型的局限性，计算机只能识别一些词汇或一些句子。语音识别系统在语种改变时，往往会出现错误的识别结果。针对上述问题，结合隐马尔可夫模型原理，在 HTK 语音处理工具箱的基础上构建了中英文特定词语音识别系统。该系统通过代码控制整个构建过程，使其在更换新的训练数据和词典后能快速生成对应的识别模型。

关键词：语音识别；隐马尔可夫模型；HTK

Specific Speech Recognition System Based on HTK

ZENG Ni, FEI Hong-Xiao, JIANG Zhen-Fei

(School of software, Central South University, Changsha 410075, China)

Abstract: Having developed about 50 years, the speech recognition (SR) technique has a wide range of applications in many fields, such as voice dialing system, digital remote control and industrial control. But the limitation of acoustic and language model is that the computer can only recognize some words or sentences. When the speech language changes, the system often gets wrong results. To address the problem above, the speech recognition system has been built on the basis of HTK as well as hidden markov model theory. Controlling the building process by code, the system can quickly generate a new recognition model when the training data and dictionary has changed.

Keywords: speech recognition; Hidden Markov model; HTK

随着信息时代的到来，计算机成为人们日常生活不可缺少的工具，但人与计算机之间的交互方式仍然以鼠标键盘为主，为了使计算机能和人更好地交互，一些基于语音识别、机器翻译、语音合成的人机交互通信系统不断出现，使语音交互系统成为人机对话的一般工具。

语音识别是机器通过识别和理解过程把人类的语音信号转变为相应的文本或命令的技术。其根本目的是研究出一种具有听觉功能的机器，这种机器能直接接受人的语音，理解人的意图，并作出相应的反应[1]。

HTK 是指隐马尔可夫模型工具箱，最初由英国剑桥大学工程系的语音视觉和机器人技术工作组开发，主要应用于语音识别，也可应用于语音合成、字符识别、DNA 排序模拟等多个领域[2,3]。就语音识别系统而言，从特征提取、HMM 编辑与训练、识别、词典管理、标记文件管理等方面，HTK 都提供了丰富的工具。

目前使用的声学模型和语言模型太过局限[3-5]，以致计算机只能识别一些词汇或一些句子。如果突然从中文转到英文，计算机就会不知如何反应，而出现错误的识别，或者用户使用了某个领域的专业术语，可能不能得到正确的识别。针对上述问题，在 HTK 的基础上考虑构建中英文特定词语音识别系统。通过代码控制整个构建过程，使其在更换新的训练数据和词典后能快速生成对应的识别模型。

1 特定词语音识别系统的设计

1.1 语音数据库的设计

在自动化、工业控制等诸多领域，人们常常使用

① 收稿时间:2010-07-15;收到修改稿时间:2010-08-25

数字 0 到 9 作为指令，数字语音相对其他单词语音有着更为广泛的应用范围，因此系统选择中文和英文的数字 0 到 9 作为识别的特定词，包括英文中 0 的第二种发音 oh 总共 21 个词。

1.2 任务语法与词典的设计

该系统属于孤立词语音识别系统，任务语法比较简单，可以定义为 (SENT-START($digit)SENT-END)。其中 SENT-START 和 SENT-END 分别表示句子的开始和结束，开始和结束一般为静音。$digit 是表示要识别的数字。

语音识别系统中把识别单元划分到音素往往比直接用单词的识别效果好。但该系统中的语音数据中英文之间有些音(如，英文的/th/、中文的/ue/)在另一种语言发音中无法找到对应，因此识别单元只做到单词级。

1.3 系统建模的流程

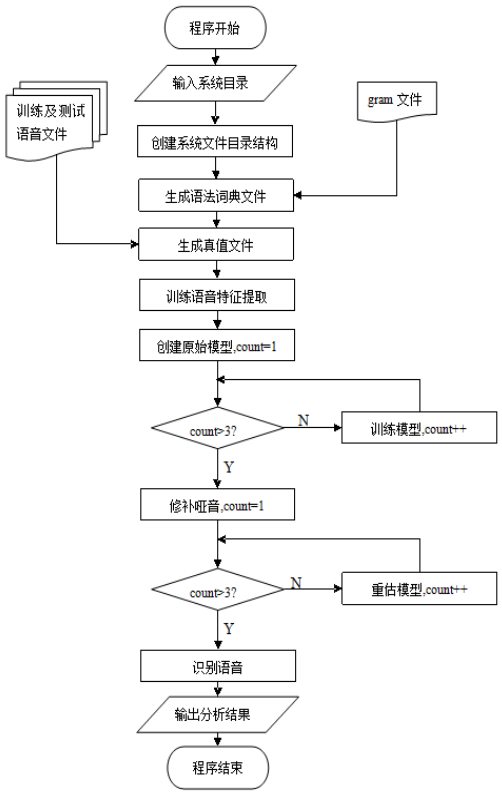

图 1　系统建模流程图

系统建模过程与一般的孤立词语音识别系统建模过程一致，其流程如图 1 所示。首先由用户给出系统所在的根目录，根目录中包含了建模过程需要用到的 gram 文件、训练及测试语音文件。其次控制代码自动生成语音识别系统的文件目录结构。根据之前给出的 gram 文件生成系统所需的语法及词典文件。根据训练语音数据和测试语音数据自动生成对应的真值文件。接着对训练语音提取特征参数。创建原始模型并将计数器置为 1。之后判断计数器是否大于 3，如果不大于则训练模型，否则进入下一步修补哑音并重置计数器到 1。重估模型三次后对测试语音进行识别，最后输出对识别结果的分析数据。

2 特定词语音识别系统的实现

2.1 训练语音的制作

所有语音数据都是使用 HTK 中的 HSLab 工具进行录制和标注。训练语音数据和测试语音数据录制好后，将所有的语音文件和标注文件存放在 data 文件夹中。data 文件夹下 的 train 文件夹中存放所有的训练语音数据，test 文件夹中存放测试语音数据。

2.2 任务语法及词典的定义

系统的任务语法比较简单，语法文件命名为 gram 存放在根目录下，值得注意的是 gram 文件结尾需要一个空行作为结束标志，否则无法被识别。gram 中的内容是任务语法的高层表示，系统无法直接使用，为了得到可用的任务语法需要用 HParse 进行转换。控制代码将自动生成 HTK 可用的任务语法文件 wdnet。

词典定义了整个系统可能出现的所有词。系统所有要识别的词有 21 个，包括静音和 sp(short pause)总共 23 个词。将这 23 个词存放在 monophones 文件中，准备另一个 monophones0 文件，内容和 monophones 基本一样，只删除了 sp，用于建立不包含 sp 的模型。词典的定义文件统一放在 dict 文件夹下。

2.3 确定特征参数并构建原始模型

系统的特征参数等配置文件存放在 def 文件夹中，提取语音数据特征参数的配置文件为 config。系统根据配置文件和特征参数目标文件自动提取参数特征。

提取完训练语音的特征参数后开始构建原始模型。构建原始模型需要：一个训练数据 mfcc 的列表，存在 trainMFCC.txt 中；配置文件 config；一个基本模型，确定模型的状态数等特征；以及要创建的模型的目录。

准备好初始化原始模型的文件后，通过控制代码在 models/hmm0 中得到生成的原始模型，模型包括两个文件 vFloors 和 proto。生成的这些值用于训练重估模型时估计的变化向量的基底。但生成的文件由于缺少 HTK 文件的头文件描述，不能直接用于模型的重估。需要控制代码修改此文件，添加头文件描述 ~o <VECSIZE> 39 <MFCC_0_D_A>。

生成的第二个文件 proto 是基本 proto 的更新。proto 文件也需要修改才能用于之后的模型训练。系统包括 sil 共有 22 个识别语音，对每个识别语音要建立 HMM 模型，每个模型的原始情况就是 proto 中数据描述的情况，因此需要将 proto 中的内容重复 21 遍(总共 22 组),每一组~h “proto”中的 proto 替换为要识别的模型名称，如 sil 对应地修改成~h “sil”。

2.4 训练模型

得到原始模型后就可以开始对模型进行训练重估。训练模型用到的 HTK 工具为 HERest，重估模型需要用到之前定义的词典、特征参数的配置文件、训练数据 MFCC 列表文件、训练数据的真值文件以及上一次训练的模型，对于第一次训练就是指 hmm0 文件夹。真值文件可以在训练语音制作的阶段制作，存放了所有训练语音数据的标注内容。

控制代码将执行三遍模型训练，分别生成 hmm1、hmm2、hmm3 三个文件夹。这三个生成的文件夹内的训练模型中没有包含 sp 的模型，在 hmm4 中加入 sp 模型。sp 类似于 sil，sp 的模型数据就来源于 sil。与 sil 模型不同的是 sp 模型使用 3 状态数，因此在复制 sil 的内容后需要修改<NumStates> 为 3，保留<State> 3 的内容并标记为<State> 2。最后复制转移矩阵的中间状态为 sp 的转移矩阵。用加入了 sp 的模型再重估三次生成 hmm5、hmm6、hmm7，hmm7 就是最终得到的训练好的模型。

2.5 识别输入语音

识别测试语音可以通过控制代码自动识别存放在 data/test 下的语音数据文件，控制代码将最后给出识别语音结果分析。另一种识别语音的方式允许用户直接输入语音，但是需要提前准备一个文件 directin.conf 配置相关参数，在控制台中输入 HVite -C def/directin.conf -e -g -H models/hmm7/hmmdefs -w wdnet dict/dict dict/monophones。出现提示符 READY[1]表示开始录音等待输入信号，按下回车可以停止录音，随后显示识别结果并播放之前录制的语音。

3 系统测试与结果分析

HTK 中的 HResults 工具可以进行系统性能评估。第一次得到对识别结果的统计数据，如图 2 所示，该系统是孤立词识别系统，因此单词识别率和句子识别率一致。

```
====================== HTK Results Analysis =============
  Date: Fri May 28 11:04:09 2010
  Ref : data/test/testwords.mlf
  Rec : results/recout.mlf
------------------------ Overall Results --------------------
SENT: %Correct=95.24 [H=20, S=1, N=21]
WORD: %Corr=95.24, Acc=95.24 [H=20, D=0, S=1, I=0, N=21]
============================================================
```

图 2 第一次 HResults 执行结果图

HResults 对 data/test/testwords.mlf 和 results/recout.mlf 两个文件进行了对比，第一个文件是应该识别的结果，后一个文件是识别输入语音时得到的识别结果。从图中可以看出单词和句子的识别率都为 95.24%，第一行 H=20 表示正确识别的数量为 20，S=1 表示识别错误的数量为 1，N=21 表示总共要识别的数量为 21。第二行的 D 表示词删除错误，I 表示词插入错误，其他的同第一行。

```
====================== HTK Results Analysis =============
  Date: Fri May 28 11:51:02 2010
  Ref : data/test/testwords.mlf
  Rec : results/recout.mlf
------------------------ Overall Results --------------------
SENT: %Correct=100.00 [H=21, S=0, N=21]
WORD: %Corr=100.00, Acc=100.00 [H=21, D=0, S=0, I=0, N=21]
============================================================
```

图 3 第二次 HResults 执行结果图

从 HResults 的结果看出有一个识别错误，检查 recout.mlf 文件，发现 tc00.rec 和 tc01.rec 都被识别成了 YI。再检查 hmm7 中 hmmdefs 里 yi 和 ling 的模型数据，对比发现两者数据一致。到存放训练数据的 data 文件夹中通过 HSLab 调出之前录制的 sig 文件，发现 ling 和 yi 的语音数据出现了重复，需要对 ling 和 yi 的训练语音进行重新录制。用新的数据建模后得到新的识别结果，如图 3 所示，执行 HResults 后得到识别率达到了 100%。

4 结论

中英文特定词语音识别系统，一方面对中英文语音同时建模，使其能够同时识别中英文的语音。但由于中英文发音本身的差异，只在单词级进行识别。另一方面，为了增加系统的扩展性以及减少模型建立的时间，通过代码控制整个系统的建模过程。控制代码能够在给出训练语音和词典的情况下建立特定词语音识别系统，并且特定词不限于中英文的单词，可以是选择的任何语种的单词，语音的识别单元都是单词级，建立的语音识别系统属于孤立词语音识别系统。

参考文献

1 易克初,田斌,付强.语音信号处理.北京:国防工业出版社,2000.191－259.

2 石现峰,张学智,张峰.基于 HTK 的语音识别系统设计.计算机技术与发展,2006,16(10):37－38.

3 Young S, Evermann G, Gales M.The HTK Book. Cambridge UniversityEngineering Department. Version 3.1, 2001.40－120.

4 孙宁,孙劲光,孙宇.基于神经网络的语音识别技术研究.计算机与数字工程,2006,34(3):58－61.

5 何湘智.语音识别的研究和发展.计算机与现代化,2002,79(3):3－6.

(上接第 148 页)

像处理速度取决于单个 cpu 核心最高速度。其余核心在程序运行期间不能得到充分的利用。

2) 多线程程序的图像处理速度取决于线程个数和核心个数的最小值。

3) 当核心个数大于线程个数的时候，图像解码的核心个数等于线程个数，图像解码的速度取决于线程个数；当线程个数大于核心个数的时候，系统将各个线程优化到每个核心里边，图像解码的速度取决于核心个数。

4) 将多线程技术应用在 JEPG2000 图像解码中，可以充分利用多核心计算机的资源，提高图像解码速度。在线程个数大于电脑核心个数的时候，双核、四核、八核的计算机，相对单核处理速度分别可以提高两倍，四倍和八倍。

参考文献

1 张困申,晓黄.Api for Windows98/2000.北京:清华大学出版社,2001.

2 Jeffery R. Programming Applications for Microsoft Windows, 1996.61－62.

3 Jim B, Robert W.Multithreading Applications in Win32, 2002.

4 Jim B, et al. 侯捷译. WIN32 多线程程序设计.武汉:华中科技大学出版社,2002.

5 ISO/IEC 1.29.15444 JPEG-2000: JPEG 2000 Part I Final Draft International Standard. Cupertino: SO/IEC, 2000.

6 Wang L.The Core Algorithm and New Developments of JPEG2000 Standard. Tracks for Standard & Technology, 2005: 67－69.

7 刘凯,李云松,吴成柯.一种比特平面并行处理的零数编码结构.电路与系统学报,2005,(5):23－24.

8 邓家先.遥感图像编码技术研究[博士学位论文].西安:西安电子科技大学,2004.