应用HTK建立连续语音识别系统

----------王风娜

基本内容

- 知识回顾

  HTK工具包

  基于HMM的连续语音识别

- 应用HTK建立连续语音识别系统实例

知识回顾

- HTK工具包

数据准备工具

HDMan、HCopy、***HLEd***、HSGen、HBuild、HLStats 、HParse

模型训练及优化工具

HERest、HInit、HRest、***HHEd***、HCompV

识别工具

HVite

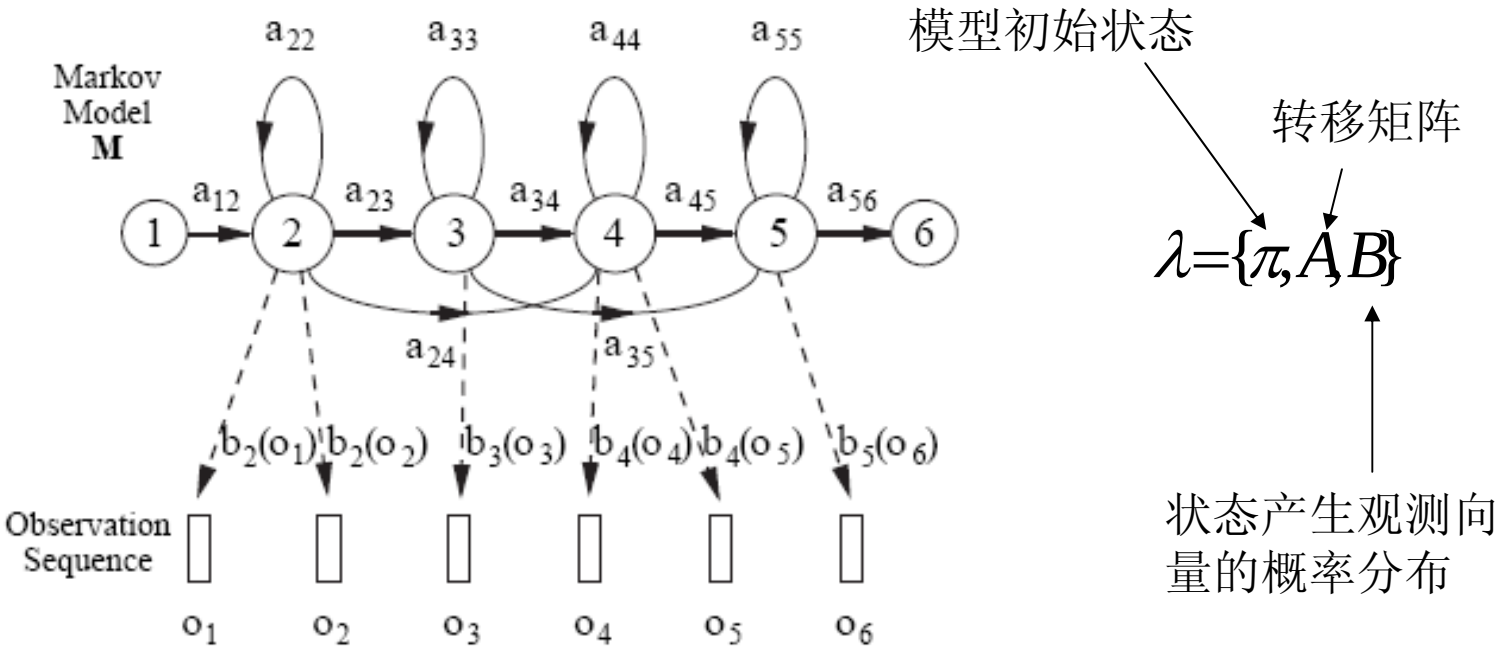
性能评估工具

HResults、HRec

基于HMM的连续语音识别系统

- HMM

三个基本问题：推理、学习、识别



连续语音识别

模型结构：混合HMM(见附)

建模单元：可根据实际问题选择，对于大词汇量选择音素，进而扩展到三音素（词内或词间）

三音素捆绑：解决训练数据不足问题

基于数据的状态聚类：自底向上，不能给不可见音素建模（欧氏距离）

基于决策树的聚类：自顶向下，能为不可见音素建模(见附)

嵌入式训练：训练语音必须有对应的抄本文件

嵌入式识别：

$$w^* = \arg\max_w p(w|O) = \arg\max_w \frac{p(w)p(O|w)}{p(O)}$$

识别网络及N-gram语言模型

识别结果评估：

$$Correct = \frac{N-D-S}{N} \times 100\%$$

$$Accuracy = \frac{N-D-S-I}{N} \times 100\%$$

应用HTK建立连续语音识别的实例

- 数据准备
- 创建模型及学习

  单音素模型
  三音素模型
  状态捆绑
  增加高斯混合模型的个数
- 识别及评估

数据准备

训练及待识别语音文件（.wav）

训练语音包含的所有词（wlist）

训练语音的词级标注文件（.lab 或 word.mlf）

wlist中词的发音词典dict(见附)

HDMan -m -w wlist -n monophones1 -l dlog dict beep names

训练语音的音素级标注文件（.lab 或 phones.mlf）

HLEd .led 生成音素序列文件phones0（不包含sp）、phones1（

训练语音的特征文件（.mfc、.plp等）

HCopy config

训练语音词级网络wdnet(见附)

HPRase HBuild HParse

```
#!MLF!#
"*/S0001.lab"
ONE
VALIDATED
ACTS
OF
SCHOOL
DISTRICTS
.
"*/S0002.lab"
TWO
OTHER
CASES
ALSO
WERE
UNDER
ADVISEMENT
.
"*/S0003.lab"
BOTH
FIGURES
(etc.)
```

创建模型及学习(逐步细化)

1、单音素模型：

Proto文件：定义模型拓扑结构

3-state left-right

HCompV：统计训练数据全局均值、方差

HCompV -C config -f 0.01 -m -S train.scp -M hmm0 proto

⟶ hmm0（marcos、hmmdef）

HERest： **X3** ⟶ hmm3

HERest -C config -I phones0.mlf -t 250.0 150.0 1000.0 -S train.scp -H hmm0/macros –H hmm0/hmmdefs -M hmm1 ***phones0***

```
~o <VecSize> 39 <MFCC_0_D_A>
~h "proto"
<BeginHMM>
 <NumStates> 5
   <State> 2
      <Mean> 39
        0.0 0.0 0.0 ...
      <Variance> 39
        1.0 1.0 1.0 ...
   <State> 3
      <Mean> 39
        0.0 0.0 0.0 ...
      <Variance> 39
        1.0 1.0 1.0 ...
   <State> 4
      <Mean> 39
        0.0 0.0 0.0 ...
      <Variance> 39
        1.0 1.0 1.0 ...
   <TransP> 5
    0.0 1.0 0.0 0.0 0.0
    0.0 0.6 0.4 0.0 0.0
    0.0 0.0 0.6 0.4 0.0
    0.0 0.0 0.0 0.7 0.3
    0.0 0.0 0.0 0.0 0.0
<EndHMM>
```

2、固定静音的单音素模型

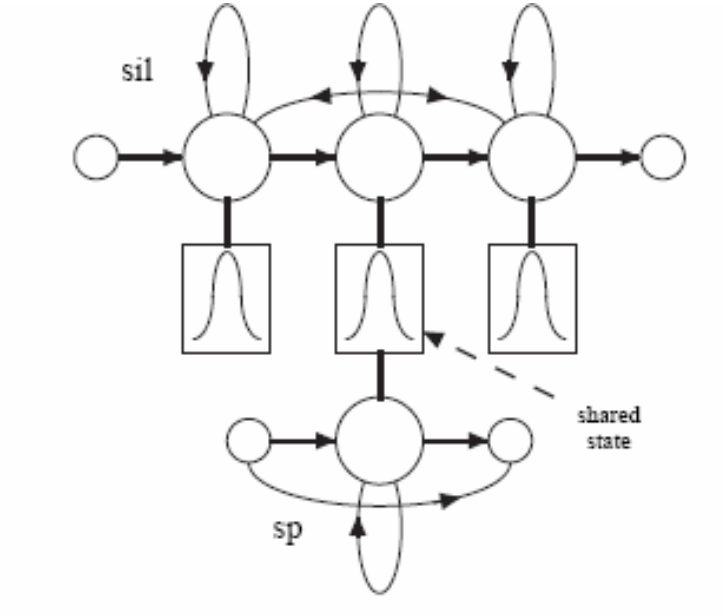
加入sil、sp模型 ⟶ hmm4

捆绑sil和sp模型 ⟶ hmm5

HHEd -H hmm4/macros -H hmm4/hmmdefs -M hmm5 sil.hed ***phones1***

HERest：

X2 ⟶ hmm7 （***phones1***）




```
AT 2 4 0.2 {sil.transP}
AT 4 2 0.2 {sil.transP}
AT 1 3 0.3 {sp.transP}
TI silst {sil.state[3],sp.state[2]}
```

3、训练数据的重组：解决多音现象

HVite：得到更好的音素级标注文件**aligned.mlf**

HVite -l '*' -o SWT -b silence -C config -a -H hmm7/macros -H hmm7/hmmdefs
-i aligned.mlf -m -t 250.0 -y lab -I words.mlf -S train.scp dict ***phones1***

HERest：

X2 ——→ hmm9（*phones1*）

Error：can't find tee model at the start and end of the sentence.

4、三音素模型

三音素标注文件wintri.mlf

```
WB sp
WB sil
TC
```

HLEd -n triphones1 -l '*' -i wintri.mlf mktri.led aligned.mlf

sil th ih s sp m ae n sp ...

扩展为

sil th+ih th-ih+s ih-s sp m+ae m-ae+n ae-n sp ...

三音素模型

HHEd -B -H hmm9/macros -H hmm9/hmmdefs -M hmm10 mktri.hed phones1

HERest： X2 ——→ hmm12（*triphones*）

-s stats

```
CL triphones1
TI T_ah {(*-ah+*,ah+*,*-ah).transP}
TI T_ax {(*-ax+*,ax+*,*-ax).transP}
TI T_ey {(*-ey+*,ey+*,*-ey).transP}
TI T_b {(*-b+*,b+*,*-b).transP}
TI T_ay {(*-ay+*,ay+*,*-ay).transP}
...
```

5、三音素捆绑

HHEd：

HHEd -B -H hmm12/macros
-H hmm12/hmmdefs
-M hmm13 tree.hed
triphones1 > log

→

HERest：

X2 hmm15（*tiedlist*）

```
RO 100.0 stats
TR 0
QS "L_Class-Stop" {p-*,b-*,t-*,d-*,k-*,g-*
QS "R_Class-Stop" {*+p,*+b,*+t,*+d,*+k,*+g
QS "L_Nasal" {m-*,n-*,ng-*}
QS "R_Nasal" {*+m,*+n,*+ng}
QS "L_Glide" {y-*,w-*}
QS "R_Glide" {*+y,*+w}
....
QS "L_w" {w-*}
QS "R_w" {*+w}
QS "L_y" {y-*}
QS "R_y" {*+y}
QS "L_z" {z-*}
QS "R_z" {*+z}
TR 2

TB 350.0 "aa_s2" {(aa, *-aa, *-aa+*, aa+*).state[2]}
TB 350.0 "ae_s2" {(ae, *-ae, *-ae+*, ae+*).state[2]}
TB 350.0 "ah_s2" {(ah, *-ah, *-ah+*, ah+*).state[2]}
TB 350.0 "uh_s2" {(uh, *-uh, *-uh+*, uh+*).state[2]}
....
TB 350.0 "y_s4" {(y, *-y, *-y+*, y+*).state[4]}
TB 350.0 "z_s4" {(z, *-z, *-z+*, z+*).state[4]}
TB 350.0 "zh_s4" {(zh, *-zh, *-zh+*, zh+*).state[4]}

TR 1

AU "fulllist"
CO "tiedlist"
ST "trees"
```

6、增加混合高斯模型个数

Increase the mixture

HHEd -H hmm15/macros -H hmm15/hmmdefs -M hmm16 increasemix.hed tailist

increasemix.hed: MU +2 {*.state[2-4].mix}

HERest

X2 ⟶ hmm18（*tiedlist*）

可根据实际问题需要适当增加高斯模型个数

识别及评估

识别HVite

HVite -H hmm15/macros -H hmm15/hmmdefs -S test.scp -l '*' -i recout.mlf -w wdnet
-p 0.0 -s 5.0 dict tiedlist

词级、音素级、三音素级识别结果

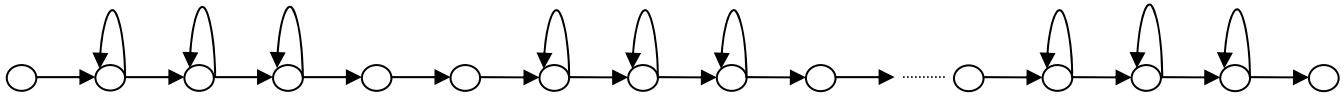
评估

HResults -I testref.mlf tiedlist recout.mlf >result

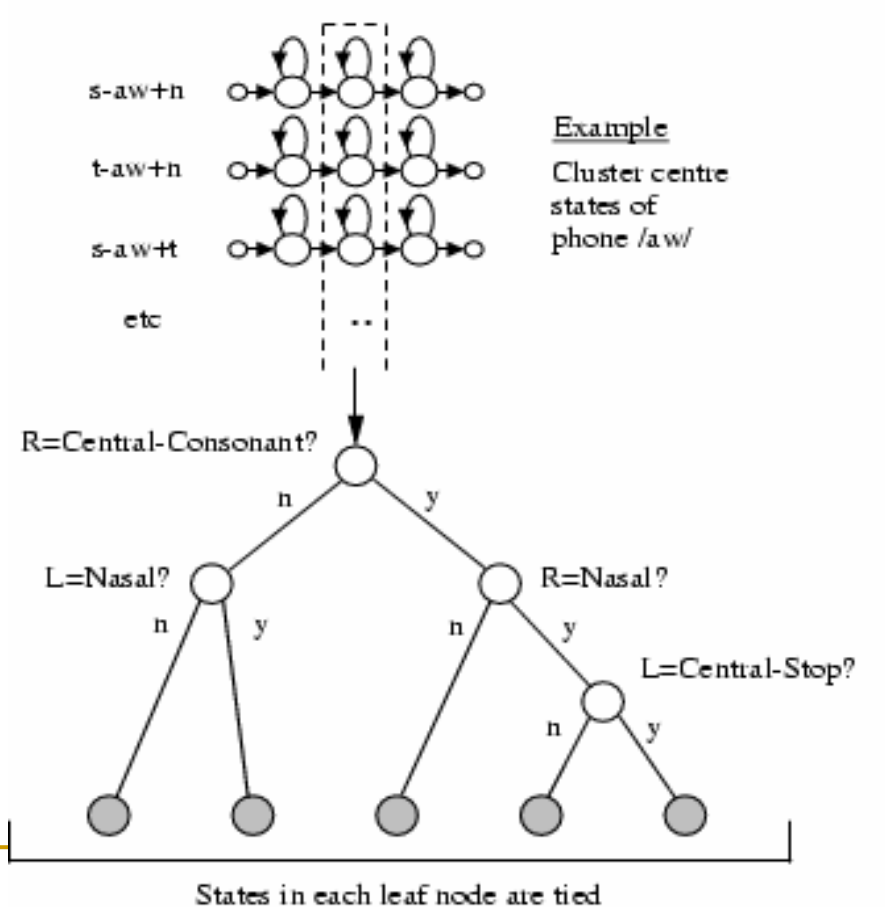
```
====================== HTK Results Analysis =============
  Date: Sun Oct 22 16:14:45 1995
  Ref : testrefs.mlf
  Rec : recout.mlf
---------------------- Overall Results ----------------
SENT: %Correct=98.50 [H=197, S=3, N=200]
WORD: %Corr=99.77, Acc=99.65 [H=853, D=1, S=1, I=1, N=855]
============================================================
```

附:

模型结构：混合HMM



决策树捆绑:



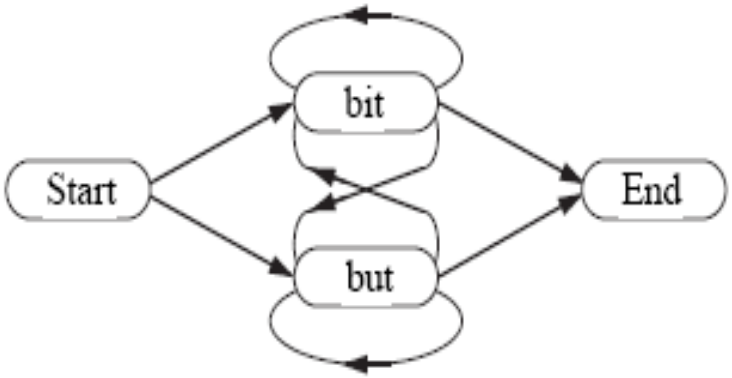
返回

词典dict

```
A                 ah sp
A                 ax sp
A                 ey sp
CALL              k ao l sp
DIAL              d ay ax l sp
EIGHT             ey t sp
PHONE             f ow n sp
SENT-END    []    sil
SENT-START  []    sil
SEVEN             s eh v n sp
TO                t ax sp
TO                t uw sp
ZERO              z ia r ow sp
```

返回

识别网络




```
# Define size of network: N=num nodes and L=num arcs
N=4 L=8
# List nodes: I=node-number, W=word
I=0 W=start
I=1 W=end
I=2 W=bit
I=3 W=but
# List arcs: J=arc-number, S=start-node, E=end-node
J=0 S=0 E=2
J=1 S=0 E=3
J=2 S=3 E=1
J=3 S=2 E=1
J=4 S=2 E=3
J=5 S=3 E=3
J=6 S=3 E=2
J=7 S=2 E=2
```

返回

The End !