Tutorial of Building an LVCSR System using HTK

Shih-Hsiang Lin(林士翔)
Department of Computer Science & Information Engineering
National Taiwan Normal University

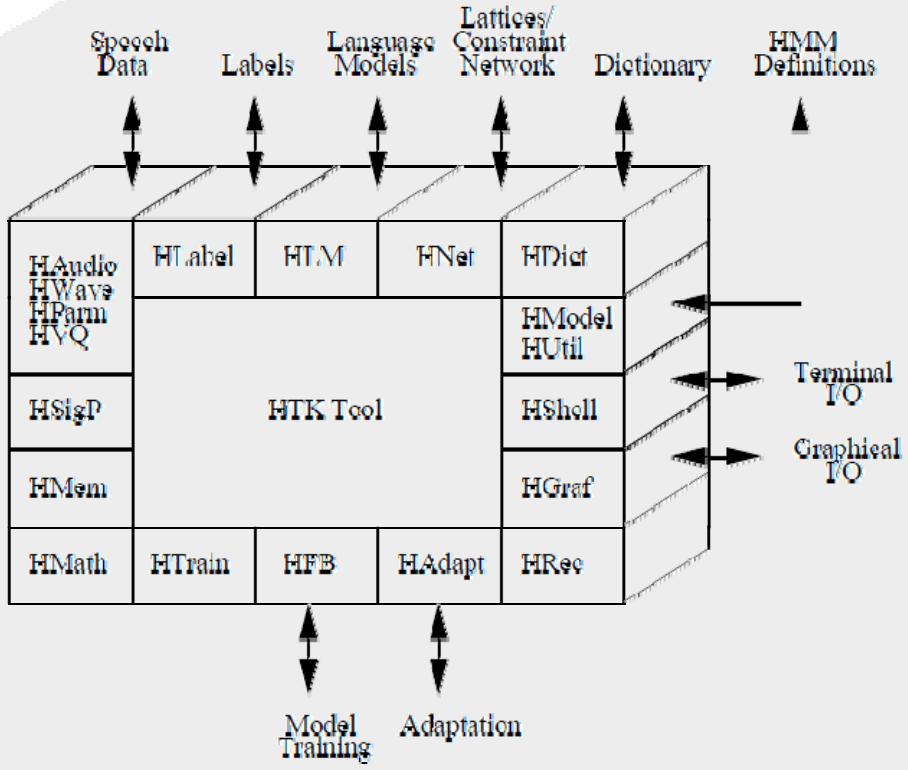
Reference:
-Steve Young et al, The HTK Books (for HTK Version 3.4), 2009

An Overview of HTK

- HTK: A toolkit for building Hidden Markov Models
- HMMs can be used to model any time series and the core of HTK is similarly general-purpose
- HTK is primarily designed for building HMM-based speech processing tools, in particular speech recognizers

An Overview of HTK (cont.)

- Generic Properties of an HTK Tools
  - HTK tools are designed to run with a traditional command line style interface



Building a LVCSR System

- Two major processing stages are involved
  - Training Phase: The training tools are used to estimate the parameters of a set of HMMs using training utterances and their associated transcriptions
  - Recognition Phase: Unknown utterances are transcribed using the HTK recognition tools

Training Phase

- Step1 - Data Preparation
- Step2- Creating Monophone HMMS
  - Creating Flat Start Monophones
  - Fixing the Silence Models
- Step3- Creating Tied-State Triphone
  - Making Triphones from Monophones
  - Making Tied-State Triphones
  - Splitting Mixture Number

Creating Monophone HMMs –
Creating Flat Start Monophones (1/3)

- Compute the global mean and variance and set all of the Gaussians in a given HMM to have the same mean and variance

```
HCompV.exe -C .\config\Config.fig  -m -S .\config\TRAINING_LIST.scp
                -M   .\Models\hmmo   .\config\pro_39_m1_s3
```

```
TARGETKIND=MFCC_0_D_A_Z
TARGETRATE=100000.0     #frameshift 10ms
WINDOWSIZE=320000.0    # framesize = 32ms
PREEMCOEF=0.97
NUMCHANS=26
CEPLIFTER=22
NUMCEPS=12
.................
.................
```

Config.fig

```
.\coeff\T00001.mfc
.\coeff\T00002.mfc
.\coeff\T00003.mfc
.\coeff\T00004.mfc
.\coeff\T00005.mfc
.\coeff\T00006.mfc
.\coeff\T00007.mfc
.\coeff\T00008.mfc
.................
.................
```

TRAINING_LIST.scp

```
<BeginHMM>
<NumStates> 3
<VecSize> 39
<MFCC_E_D_A_Z> <nullD> <diagC>
<StreamInfo> 1 39
<State> 2
<NumMixes> 1
<Stream> 1
<Mean> 39
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
<Variance> 39
1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
<TransP> 3
0.000e+0   1.000e+0   0.000e+0
0.000e+0   7.000e-1   3.000e-1
0.000e+0   0.000e+0   0.000e+0
<EndHMM>
```

pro_39_m1_s3

Creating Monophone HMMs –
Creating Flat Start Monophones (2/3)

- Created a Master Macro File(MMF) called hmmdefs containing a copy for each of the required monophone HMM
  - constructed by manually copying the prototype and relabeling it for each required monophone

```
copy   .\Models\hmmo\pro_39_m1_s3  .\Models\BasicPhoneSet\@
.....
copy   .\Models\hmmo\pro_39_m1_s3  .\Models\BasicPhoneSet\y
MakeHMMDef.exe .\Config\MonoPhoneSet .\Models\BasicPhoneSet .\Models\hmmo\hmmdefs
```

```
@
N'
U'
U`
Z`
a
B
.....
....
```

MonoPhoneSet

Creating Monophone HMMs –
Creating Flat Start Monophones (cont.) (3/3)

- Manually edit the resultant prototype HMMs in the directory [hmmo\hmmdefs] to remove the row ~h "pro_39_m1_s3"
- The flat start monophones stored in the directory hmmo are re-estimated using the embedded re-estimation tool HERest

```
HERest -C  .\config\Config.fig -I .\config\labels\TRANSCRIPTION_PHONE_NO_SP.mlf
        -X lab -t 250.0 150.0 1000.0 -S .\config\scp\TRAINING_LIST.scp
        -H .\Models\hmmo\pro_39_m1_s3    -H .\Models\hmmo\hmmdefs
        -M  .\Models\hmm1  .\Config\MonoPhoneSet
```

```
#!MLF!#
"*/Tooo01.lab"
sil
d
i
N'b
@
nsj
l
.....
.....
```

TRANSCRIPTION_PHONE_NO_SP.mlf

You can repeat the above command multiple times, e.g., 5 times to achieve a better set of monophone HMMs

Creating Monophone HMMs – Fixing the Silence Models (1/3)

- Use a text editor on the file hmm5/hmmdefs to copy the centre state of the sil model to make a new sp model and store the resulting MMF hmmdefs, which includes the new sp model, in the new directory hmm6

```
~h "sp"
<BEGINHMM>
<NUMSTATES> 3
<STATE> 2
<MEAN> 39
-5.009696e+000 2.551631e+000 -2.409606e+000 8.226590e+000 7.124836e+000 2.665814e+000 6.869464e+000 3.271267e+000 3.952755e+000 2.974891e+00
1.218464e-001 -1.765260e-001 -9.396194e-002 -7.255821e-002 -1.159361e-001 3.830089e-001 1.512990e-001 7.003973e-002 1.734397e-001 -2.435962e-001 -2.5
<VARIANCE> 39 2.384063e+001 2.230153e+001 3.703096e+001 3.906878e+001 3.690832e+001 3.838507e+001 3.178211e+001 2.957892e+001 2.980945e+001 3
2.494909e+000 2.339501e+000 2.525849e+000 2.338489e+000 2.217574e+000 2.031253e+000 2.070557e+000 3.007780e-001 2.547940e-001 4.294375e-001 4.4
<GCONST> 1.136243e+002
<TRANSP> 3
0.000000e+000 1.000000e+000 0.000000e+000
0.000000e+000 4.994126e-001 5.005874e-001
0.000000e+000 0.000000e+000 0.000000e+000
<ENDHMM>
```

Creating Monophone HMMs –
Fixing the Silence Models (cont.) (2/3)

- Run the HMM editor HHEd to add the extra transitions required and tie the sp state to the centre sil state

```
HHEd -H .\Models\hmm6\pro_39_m1_s3  -H .\Models\hmm6\hmmdefs
     -M .\Models\hmm7  .\config\sil.hed .\config\MonoPhoneSetSP
```

Add a transition from state 2 to state 4 with probability 0.2

```
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]}
```

Create a tied-state call silst

sil.hed

Creating Monophone HMMs – Fixing the Silence Models (cont.) (3/3)

- Finally, another passes of HEREST are applied using the phone transcriptions with sp models between words

```
HERest -C .\config\Config.fig  -I .\config\labels\TRANSCRIPTION_PHONE.mlf
        -X lab -t 250.0 150.0 1000.0 -S .\config\scp\TRAINING_LIST.scp
        -H .\Models\hmm7\pro_39_m1_s3    -H .\Models\hmm7\hmmdefs
        -M  .\Models\hmm8  .\Config\MonoPhoneSetSP
```

```
#!MLF!#
"*/Tooo01.lab"
sil
d
i
N'
sp
b
@
n
sp
.................
.................
```

TRANSCRIPTION_PHONE.mlf

Creating Tied-State Triphone–
Making Triphones from Monophones (1/3)

- Convert the monophone transcriptions in TRANSCRIPTION_PHONE.mlf to an equivalent set of triphone transcriptions in wintri.mlf.

```
HLEd -n triphones1 -l *  -i .\Config\labels\wintri.mlf -X lab  .\Config\mktri.led
     .\Config\labels\TRANSCRIPTION_PHONE.mlf
```

Convert all phoneme labels to triphones

TC

mktri.led

```
#!MLF!#
"*/Tooo01.lab"
sil
d
i
N'
sp
b
@
n
sp
.....
.....
```

TRANSCRIPTION_PHONE.mlf

```
#!MLF!#
"*/Tooo01.lab"
sil
sil-d+i
d-i+N`
i-N'+sp
N'-sp+b
sp-b+@
b-@+n
@-n+sp
n-sp+sj
.....
.....
```

wintri.mlf

```
sil
sil-d+I
d-i+N`
i-N'+sp
N'-sp+b
sp-b+@
b-@+n
@-n+sp
n-sp+sj
sp-sj+I
sj-i+a
.....
.....
```

triphones1

Creating Tied-State Triphone–
Making Triphones from Monophones (2/3)

- Cloning triphone models from monophones

```
HHEd  -H .\Models\hmm10\pro_39_m1_s3  -H .\Models\hmm10\hmmdefs
      -M .\Models\hmm11  .\Config\mktri.hed .\Config\MonoPhoneSetSP
```

Clone a HMM list

```
CL triphones1
 TI T_@          {(*-@,@+*,*-@+*).transP}
 TI T_N`         {(*-N',N'+*,*-N'+*).transP}
 TI T_U'         {(*-U',U'+*,*-U'+*).transP}
 TI T_U`         {(*-U`,U`+*,*-U`+*).transP}
 TI T_Z`         {(*-Z`,Z`+*,*-Z`+*).transP}
 TI T_a          {(*-a,a+*,*-a+*).transP}
 TI T_b          {(*-b,b+*,*-b+*).transP}
 TI T_d          {(*-d,d+*,*-d+*).transP}
 TI T_dj         {(*-dj,dj+*,*-dj+*).transP}
 .................
 .................
```

mktri.hed

Creating Tied-State Triphone–
Making Triphones from Monophones (3/3)

- Once the context-dependent models have been cloned, the new triphone set can be re-estimated using HERest

```
HERest -C .\config\Config.fig  -I .\Config\labels\wintri.mlf  -X lab -t 250.0 150.0 1000.0
        -S .\Config\scp\TRAINING_LIST.scp  -H .\Models\hmm11\pro_39_m1_s3
        -H .\Models\hmm11\hmmdefs   -M  .\Models\hmm12  triphones1
```

  - You also can repeat the above command multiple times

**For the final pass of HERest, the -s option should be used to generate a file of state occupation statistics called stats for latter used

Creating Tied-State Triphone–
Making Tied-State Triphone (1/2)

- Tie states within triphone sets in order to share data and thus be able to make robust parameter estimates.
  - HTK provides two mechanisms which allow states to be clustered and then each cluster tied
    - The first is data-driven and uses a similarity measure between states
    - The second uses decision trees and is based on asking questions about the left and right contexts of each triphone (🡸 use this)

```
HHEd -H .\Models\hmm16\pro_39_m1_s3  -H .\Models\hmm16\hmmdefs
     -M .\Models\hmm17  .\Config\tree.hed triphones1 > log
```

Load the statistics file generated at the end of the previous step and remove outlier state

Define questions

Cluster one specific set

```
RO 100.0 stats
TR 0
QS              "L_G_yuanyin"  { a-*,o-*,@-*,e-*,i-*,u-*,y-*,U'-*,U`-*,m-*,n-*,l-*,Z`-*,N'-* }
QS              "L_G_tufa"     { b-*,p-*,d-*,t-*,g-*,k-*,dz-*,ts-*,dz`-*,ts`-*,dj-*,tj-* }
.................
TR 1
TB 350.0 "ST_b_2_" {("b","*-b+*","b+*","*-b").state[2]}
TB 350.0 "ST_p_2_" {("p","*-p+*","p+*","*-p").state[2]}
.................
AU "fulllist"
CO "tiedlist"
ST "trees"
```

tree.hed

Creating Tied-State Triphone–
Making Tied-State Triphone(2/2)

- After model tying, we again repeat the following command N times

```
HERest -C .\config\Config.fig  -I .\Config\labels\wintri.mlf  -X lab -t 250.0 150.0 1000.0
        -S .\Config\scp\TRAINING_LIST.scp   -H .\Models\hmm17\pro_39_m1_s3
        -H .\Models\hmm17\hmmdefs   -M  .\Models\hmm18  tiedlist
```

```
p-ts+a
p-ts+b
p-ts+d
p-ts+e p-ts+d
p-ts+f p-ts+b
p-ts+g
p-ts+h p-ts+g
p-ts+i p-ts+d
.....
.....
```

tiedlist

Creating Tied-State Triphone–

Triphone State Mixture Split (1/2)

- Split the single Gaussian distribution of each HMM state into N mixture of Gaussian distributions, while the mixture number is set with respect to size of the training data for each model

```
HHEd  -H  .\Models\hmm25\pro_39_m1_s3  -H .\Models\hmm25\hmmdefs
      -M  .\Models\hmm26  .\Config\split.hed  .\tiedlist
```

Increase the mixture number to 4

```
MU  4 { @-sj+y.state[2-4].mix }
MU  2 { @-t+a.state[2-4].mix }
MU  4 { N'-m+e.state[2-4].mix }
MU  1 { N'-o+Z`.state[2-4].mix }
MU  8 { N'-s+U'.state[2-4].mix }
.................
.................
```

split.hed

Creating Tied-State Triphone–
Triphone State Mixture Split (2/2)

- Finally, these modes are re-estimated *N* times

```
HERest -C .\config\Config.fig  -I .\Config\labels\wintri.mlf  -X lab -t 250.0 150.0 1000.0
        -S .\Config\scp\TRAINING_LIST.scp   -H .\Models\hmm26\pro_39_m1_s3
        -H .\Models\hmm26\hmmdefs   -M  .\Models\hmm27  tiedlist
```

HTK Large Vocabulary Decoder (HDecode)

- HDecode has been specifically written for ASR tasks using cross-word triphone models

- Known restrictions are
  - only works for cross-word triphones
  - supports *N*-gram language models up to tri-grams
  - *sil* and *sp* models are reserved as silence models

HTK Large Vocabulary Decoder (HDecode) (cont.)

Pruning beam width

Word insertion penalty

```
HDecode.exe  -C config\config.fig -o ST -t 100.0 -H Models\hmm69\pro_39_m1_s3
             -H Models\hmm69\hmmdefs -S scp\Test.scp -l TEST_REC\ -p -20.0 -s 5.0
             -w config\LM2008.wid.lm config\NTNULexicon2008-72k_Wid .\tiedlist
```

LM Scale factor

```
\data\
ngram 1=72650
ngram 2=955799
\1-grams:
-0.8307618      </s>
-99             <s>             -1.393977
-3.691592       A0              -0.9617367
-2.594038       A1              -1.722824
-7.654071       A10
-5.397256       A100            -0.2698582
-7.654071       A1000
-6.932972       A10000
-3.878742       A10001          -0.5485143
.................
.................
```

LM2008.wid.lm

```
<s> sil
</s> sil
!!UNK sil
!ENTER sil
!EXIT sil
A0 b a
A1 b a
A2 b a
A3 b a
A4 b a
A5 b a
A6 b a
A7 b a
.................
.................
```

NTNULexicon2008-72k_Wid

HTK Large Vocabulary Decoder (HDecode) (cont.)

```
Convert_WID_to_Word_Char.exe Config\Dict_WID_Map Config\TestList.txt REC TEST_REC
                                                  TEST_REC_WORD TEST_REC_CHAR

NResults.exe  -L REF_REC\REF_CHAR rec -D TEST_REC_CHAR rec -S config\TestList.txt
              -P Results_Char.txt  >> Results1.txt
```

```
------------------------ Overall Results --------------------------
SENT: %Correct=0.00 [H=0, S=50, N=50]
WORD: %Corr=67.66, Acc=66.36 [H=1617, D=48, S=725, I=31, N=2390]
================================================================
```

Homework

- Try to improve the recognition rate
  - Different times of model training
  - Different numbers of mixture splitting
  - Different parameter settings
    - scale factor, word insertion penalty, pruning beam width
  - Different LM order