HTK guide

- 2015-6-10 written by JK

- 2015-6-17 edited by JK

1. Data Preparation

(1) create word list

* wordlist.txt (eojeol)

(2) create dictionary

* two versions:
* all\_dict\_no\_variation.txt
* all\_dict\_no\_variation\_sp.txt (sp inserted)

(3) create MLF

* 1. word level MLF
* kor\_word.mlf (Korean)
* word1\_5.mlf (Romanized)
* 2. phone level MLF
* phone1\_5.mlf
* phone1\_5sp.mlf
* prepare two files:
* mkphones0.led
* mkphones1.led (for sp training)

(4) coding the data

: to parameterise the raw speech waveforms into sequences of feature vectors.

* config\_param: configuration parameters included
* trainingcode.scp:   
  e.g.   
  /Users/jaegukang/kimchi/wav/fv01\_t01\_s01.wav /Users/jaegukang /kimchi/mfc/fv01\_t01\_s01.mfc

/Users/jaegukang/kimchi /wav/fv01\_t01\_s02.wav /Users/jaegukang /kimchi/mfc/fv01\_t01\_s02.mfc

/Users/jaegukang/kimchi /wav/fv01\_t01\_s03.wav /Users/jaegukang /kimchi/mfc/fv01\_t01\_s03.mfc

/Users/jaegukang/kimchi /wav/fv01\_t01\_s04.wav /Users/jaegukang /kimchi/mfc/fv01\_t01\_s04.mfc

…

2. Creating Monophone HMMs

(1) create flat start monophones (initializing HMMs)

* edit trainingcoding.scp with the appropriate path first

**> HCopy -T 1 -C ./config/config\_param -S ./scp/traincoding.scp**

**> mkdir ./models/hmm0**

* edit train.scp with the appropriate path first!

**> HCompV -T 1 -C ./config/config2 -f 0.01 -S ./scp/train.scp -M ./models/hmm0 ./config/proto**

* config2 already includes ‘TARGETKIND = MFCC\_0\_D\_A’ option for 3 versions of MFCC (original, delta, acceleration)
* copy proto to create each phone model  
  **> cd models/hmm0  
  > for i in $(cat ~/kimchi/list/phone.list); do ~/kimchi/bin/proto2eachhmm.pl $i ~/kimchi/models/HMM0/proto > $i ; done**
* create an MMF (Master Macro file. creating hmmdefs)  
  **> cd ~/kimchi  
  > ./exe/master\_macro\_mmf.sh ~/kimchi/**
* create macros file using java script  
  **> cd java; java GenerateMacros ../models/hmm0/vFloors**
* move macros in java to hmm0 folder  
  **> mv ./macros ../models/hmm0/**
* create hmm1/hmm2/hmm3 in models folder (for 3 iterations)  
  **> cd ../models; mkdir hmm1; mkdir hmm2; mkdir hmm3**
* re-train with HERest (for 3 iterations)  
  **> cd ..  
  > ./exe/reestimation\_herest.sh ~/kimchi/**

(2) fix the silence models

* check & prepare sil.hed in config folder
* open and copy the latest hmmdefs and macros into hmm4   
  **> cd models; mkdir hmm4**

**> ditto -V ./hmm3/ ./hmm4/**

* copy sil model at the end of hmmdefs with ‘sp’ as a label   
  **> cd ~/kimchi/java; java Add\_SP ../models/hmm3/hmmdefs ../models/hmm4/hmmdefs  
  > cd ~/kimchi ; mkdir ./models/hmm5; ditto -V ./models/hmm4/ ./models/hmm5/**\*To prevent any confusion, hmm4 will be simply copied to hmm5. Folder numbers correspond to those in htkbook.pdf.
* 2 more iterations  
  create hmm6/hmm7 in models folder  
  **> cd ~/kimchi/models; mkdir hmm6; mkdir hmm7**  
  **> cd ..**  
  open reestimation\_herest\_sp.sh and change line 5 as for i in 6 7 **> ./exe/reestimation\_herest\_sp.sh ~/kimchi/**

<<<<<You may skip below due to the incorrect sp modeling from htkbook.pdf. For sp modeling use the above method using java script to generate updated hmmdefs>>>>>

\*\*Problem: The sp modeling procedure below defines sp interval as 0 sec, which eventually overlaps previous labels and even doesn’t recognize proper pause between eojeols.

* open and copy the latest hmmdefs and macros into hmm4   
  **> cd models; mkdir hmm4**

**> ditto -V ./hmm3/ ./hmm4/**

* copy sil model at the end of hmmdefs with ‘sp’ as a label (MUST use TextWrangler to edit)
* open hmmdefs in hmm4
* copy and paste the “sil” model and rename the new one “sp”(don't delete your old "sil" model, you will need it - just make a copy of it)
* remove state 2 and 4 from new “sp” model (i.e. keep 'centre state' of old “sil” model in new “sp” model)
* change <NUMSTATES> to 3
* change <STATE> to 2
* change <TRANSP> to 3
* change matrix in <TRANSP> to 3 by 3 array  
  i.e.  
  0.000000e+00 1.000000e+00 0.000000e+00

0.000000e+00 6.456054e-01 3.543946e-01

0.000000e+00 0.000000e+00 9.844116e-01

* prepare for an edit script called sil.hed
* Insert sp in your phone list (list/phone.list, phonesp.list)
* generate sp using HHEd  
  > **mkdir hmm5**> **HHEd -T 1 -H ./hmm4/macros -H ./hmm4/hmmdefs -M hmm5 ../config/sil.hed ../list/phonesp.list**
* 2 more iterations  
  create hmm6/hmm7 in models folder  
  **> cd ~/kimchi/models; mkdir hmm6; mkdir hmm7**  
  **> cd ..**

open reestimation\_herest\_sp.sh and change line 5 as for i in 6 7 **> ./exe/reestimation\_herest\_sp.sh ~/kimchi/**

(3) realigning the training data (in case of multiple pronunciations)

* It’s highly recommended to open HVIte\_log file to check if any sound file is skipped. If it is, then change -t parameter values not to miss out any wav files (default. -t 250.0 150.0 1000.0)

> **HVite -A -D -T 1 -l ’\*’ -o SWT -b sil -C ./config/config2 -a -H ./models/hmm7/macros -H ./models/hmm7/hmmdefs -i ./mlf/aligned.mlf -m -t 0.0 150.0 1000.0 -y lab -I ./mlf/word1\_5.mlf -S ./scp/train.scp ./dict/all\_dict\_no\_variation\_sp.txt ./list/phonesp.list> ./log/HVite\_log**\*For triphone model, you may have to pay great amount of attention to HVite\_log. All training sound files MUST be processed without skipping any.

* two more iterations:  
  > **mkdir ./models/hmm8; mkdir ./models/hmm9**open reestimation\_herest\_sp.sh and change line 5 as for i in 8 9

> **./exe/reestimation\_herest\_sp.sh ~/kimchi/**

(4) build mixture model

* put the latest hmmdefs and macros in HMM1.2 folder  
  **> cd models; mkdir HMM1.2; ditto -V ./hmm9/ ./HMM1.2/  
  > cd ..  
  > ./exe/mixupHHEd.sh ~/kimchi/ 2** (mixture number should start from 2, 3, 4 … gradually)
* re-estimate   
  **> ./exe/reestimation\_herest\_sp\_mixup.sh ~/kimchi/ 2** (with the same mixture number!)

3. Evaluation

* prepare wav files for test in testwav folder
* edit smalltest\_coding.scp with the appropriate path and file names first!

**> HCopy -T 1 -C ./config/config\_param -S ./scp/smalltest\_coding.scp**

* execute smalltest.scp to check whether the result gets better and to decide when to stop increasing mixture number  
  > **./exe/small\_evaluate.sh ~/kimchi/ 2**
* calculate the interim result  
  > **./exe/results\_calc.sh ~/kimchi/ 2**
* check correctness of both SENT and WORD
* increase mixture number (3, 4, 5 …) gradually, check the result and stop when no more improvements found (For practice, iterate up to mixture number 7; avoid local maxima)

4. Forced Alignment

* check reliability of your trained model (e.g. fv01\_t01\_s01.wav)  
  **> ./exe/align.sh ~/kimchi/ fv01\_t01\_s01.wav** **2** (change mixture number to the latest)
* open fv01\_t01\_s01.wav and fv01\_t01\_s01.lab (created under esps folder) on Praat
* To open .lab file on Praats
* select Open
* Read from special tier file…
* Read InterverTier from Xwave…
* compare sound and textgrid