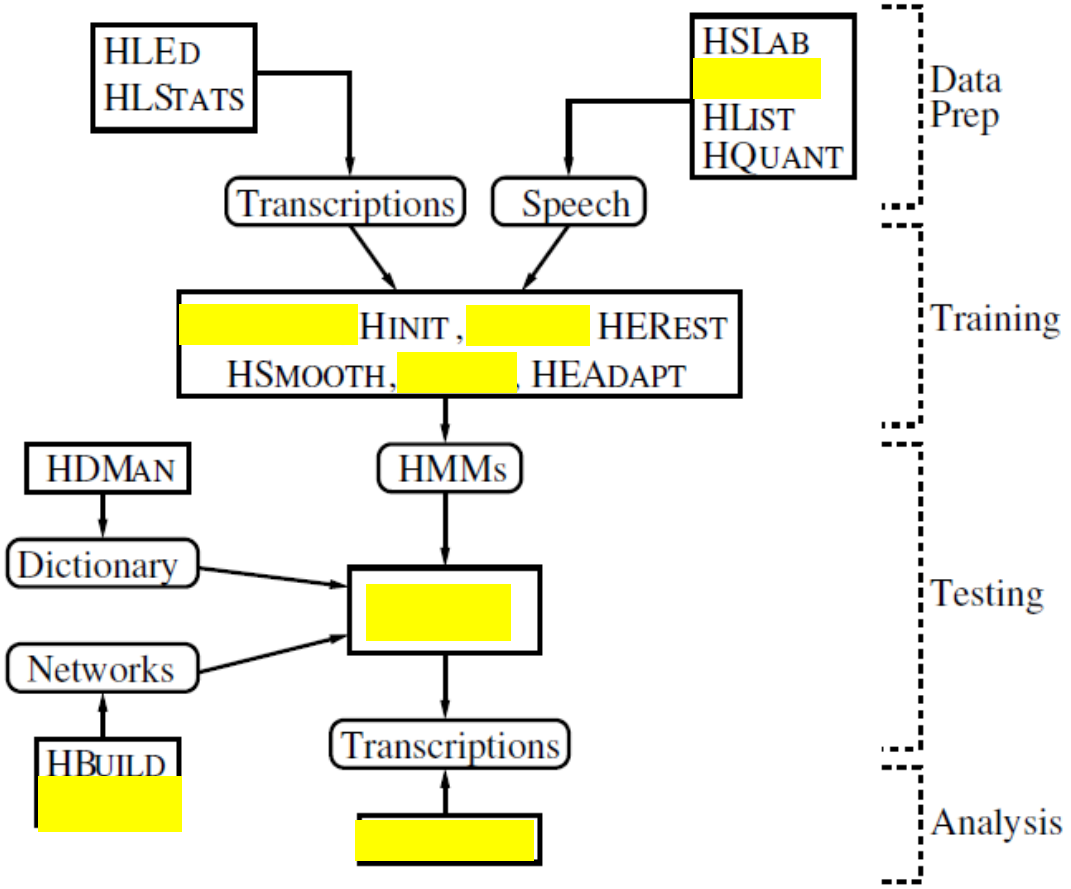
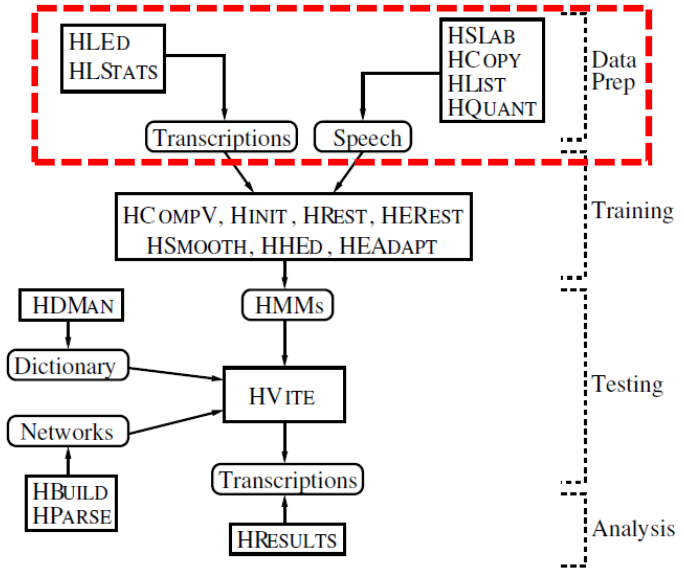
作業14
HTK 數字辨識實驗

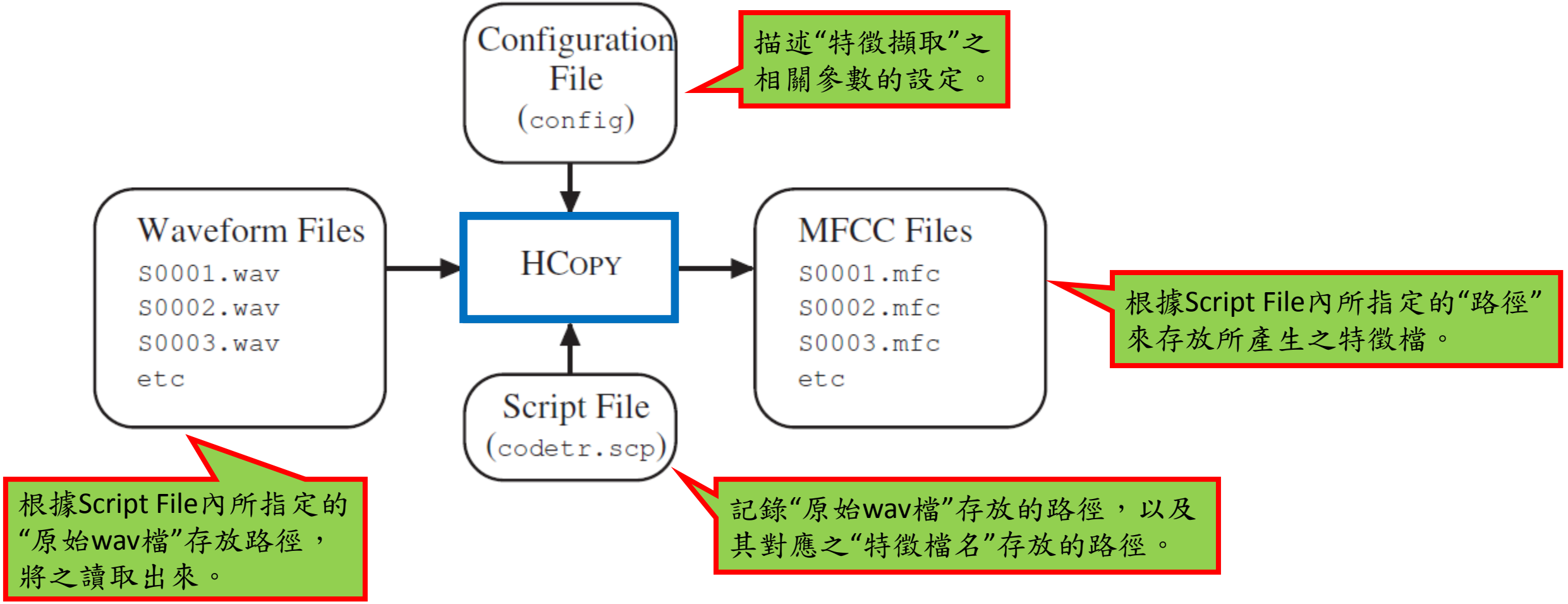
HTK Flowchart



Data Preparation



Data Prep: Feature Extraction (Using **Hcopy**)



```
HCopy [options] sa1 [ + sa2 + ... ] ta  [ sb1 [ + sb2 + ... ] tb ... ]
```

This causes the contents of the one or more source files `sa1`, `sa2`, ... to be concatenated and the result copied to the given target file `ta`.

Standard Option	Meaning
`-A`	Print command line arguments
`-B`	Store output HMM macro files in binary
`-C cf`	Configuration file is cf
`-D`	Display configuration variables
`-E dir [ext]`	Search for parent transform macros in directory dir
`-F fmt`	Set source data file format to fmt
`-G fmt`	Set source label file format to fmt
`-H mmf`	Load HMM macro file mmf
`-I mlf`	Load master label file mlf
`-J dir [ext]`	Search for transform macros in directory dir
`-K dir [ext]`	Save transform models in directory dir
`-L dir`	Look for label files in directory dir
`-M dir`	Store output HMM macro files in directory dir
`-O fmt`	Set output data file format to fmt
`-P fmt`	Set output label file format to fmt
`-Q`	Print command summary info
`-S scp`	Use command line script file scp
`-T N`	Set trace level to N
`-V`	Print version information
`-X ext`	Set label file extension to ext

Table. 4.3 Summary of Standard Options

options是大小寫有區別的，例如：-X 與 –x 不同

基本上，HTK不限定副檔名，可自定(如： .cfg, .scp)

Feature Extraction – Example (1/2)

Configuration file　　Script file

➢HCopy –C lib/hcopy.cfg -S scripts/training_hcopy.scp

(更多HCopy說明：HTK book 18.4.2)

- HCopy指令會把在Script File (**training_hcopy.scp**) 內所指定的.wav檔，進行MFCC的特徵擷取。
- Configuration File (**hcopy.cfg**)，部分參數設定說明：

```
#Coding parameters
SOURCEFORMAT=WAV
TARGETKIND=MFCC_Z_E_D_A
```

→ Input 格式為WAV檔
→ 指定特徵為：MFCC + Normalisations + Energy + Deltas + Acceleration

- **Z_** tells HTK to perform feature mean and variance (if enabled) **normalisations (i.e., zero mean and unit variance**, as described in section 5.6 and 5.10 of HTK book.
- E_D_A means that **energy**, **delta** and **acceleration** coefficients are to be computed and appended to the static MFCC coefficients.

- 關於HCopy 的Configuration File相關設定可參考HTK book (5.2 Speech Signal Processing)

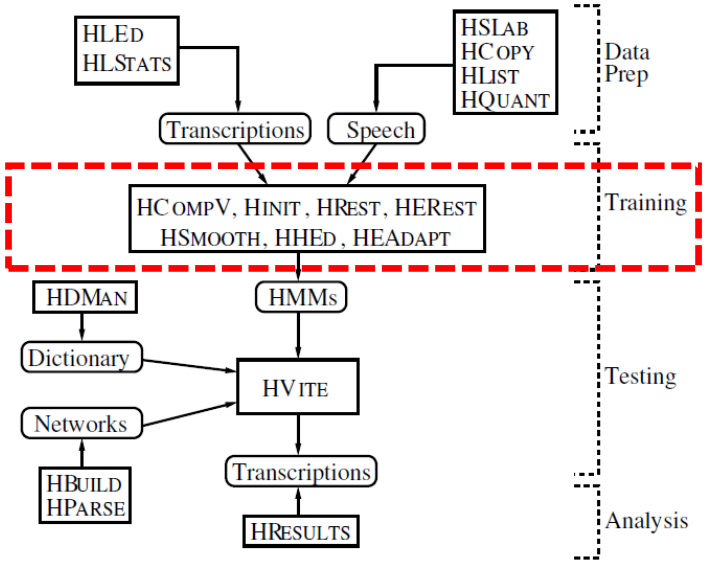
Feature Extraction – Example (2/2)

- Script File之範例 (**training_hcopy.scp**) 說明：

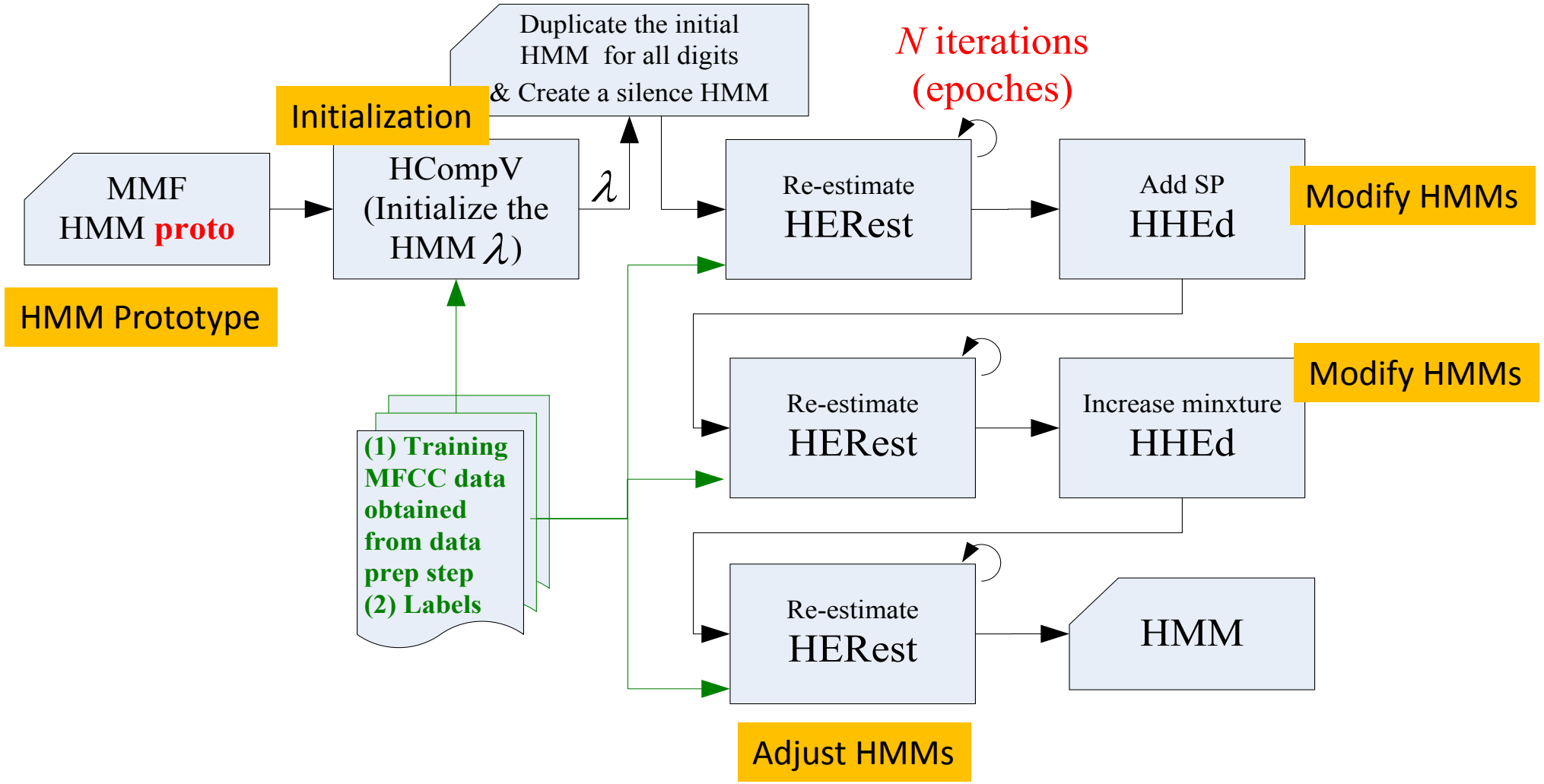
原始音訊路徑　　MFCC存放的路徑

```
speechdata/training/N110125.wav MFCC/training/N110125.mfc
speechdata/training/N110126.wav MFCC/training/N110126.mfc
speechdata/training/N110127.wav MFCC/training/N110127.mfc
speechdata/training/N110130.wav MFCC/training/N110130.mfc
speechdata/training/N110133.wav MFCC/training/N110133.mfc
speechdata/training/N110135.wav MFCC/training/N110135.mfc
```

Training


HLED
HLSTATS
HSLAB
HCOPY
HLIST
HQUANT
Data
Prep
Transcriptions
Speech
HCOMPV, HINIT, HREST, HEREST
HSMOOTH, HHED, HEADAPT
Training
HDMAN
HMMs
Dictionary
HVITE
Testing
Networks
HBUILD
HPARSE
Transcriptions
HRESULTS
Analysis

Training Flowchart



每個training data (例如N110125.mfc)都是一連串的數字發音，所以相鄰數字間會有short pause。

training.scp之範例說明：

```
MFCC/training/N110125.mfc
MFCC/training/N110126.mfc
MFCC/training/N110127.mfc
MFCC/training/N110130.mfc
MFCC/training/N110133.mfc
MFCC/training/N110135.mfc
```

HCompV - Initialization

➢HCompV -C lib/config.fig -o hmmdef -M hmm -S scripts/training.scp lib/proto

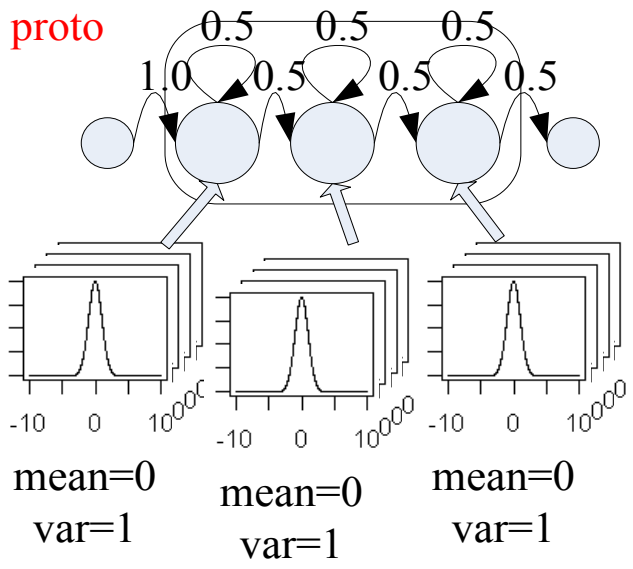
- HCompV指令說明：若是HMM-GMM，則HCompV指令會對輸入的所有特徵資料，計算其mean向量 與 co-variance矩陣。
  (可參考HTK BOOK 8.3 Flat Starting with HCompV)

- HCompV指令的參數說明:
  - -C lib/config.fig (即Configuration File；輸入之特徵資料的設定是存放在lib/config.cfg)
  - -o hmmdef -M hmm
    (HCompV 的**輸出檔名為**hmmdef ，且存放到hmm **資料夾**；
    -M hmm指明：輸出路經為hmm 資料夾； -o hmmdef指明：輸出檔名為hmmdef)
  - -S scripts/training.scp (告知HCompV：所輸入之training data (即前一步所得到的特徵) 的存放位置)
  - lib/proto (proto 為一檔案，其內容為 **HMM 模型**所需之參數定義，及其初始數值)

Initial MMF (Master Macro File) — HMM Prototype

★ **修改1**（修改<NumStates>及對應之<State>, <Variance>, <Mean>）

- proto內容說明　(可參考HTK book 7.3節)



在這proto例子裏，每個state都只有一個39維的Gaussian (mixture component數為1)，在後面的練習中會增加Mixture Gaussian的數目來提高HMM的精確度。
(改變GMM數目時，proto不必更改。)

- 註：HTK之HMM的state總數是包含開始state與結束state。

proto 存放在/lib資料夾，開檔案見完整內容。

```
~o <VECSIZE> 39 <MFCC_Z_E_D_A>
~h "proto"
<BeginHMM>
<NumStates> 5          → 共有5個states, 包含entry與exit
<State> 2              → 第2個state之參數的設定
<Mean> 39              → 特徵空間共39維度
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 ...   → 只要給定39個variances, 即co-variance matrix 的對角元素
<State> 3              → 第3個state之參數的設定
<Mean> 39
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 ...
<Variance> 39
<State> 4              → 第3個state之參數的設定
<Mean> 39
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 ...
<Variance> 39
<TransP> 5             → Transition probability matrix
0.0 1.0 0.0 0.0 0.0
0.0 0.5 0.5 0.0 0.0
0.0 0.0 0.5 0.5 0.0
0.0 0.0 0.0 0.5 0.5
0.0 0.0 0.0 0.0 0.0
<EndHMM>
```

在proto檔案內, 不需指定initial state probability vector π, 因為總是由第一個state開始。

proto檔內容

Duplicate the Initial HMM λ

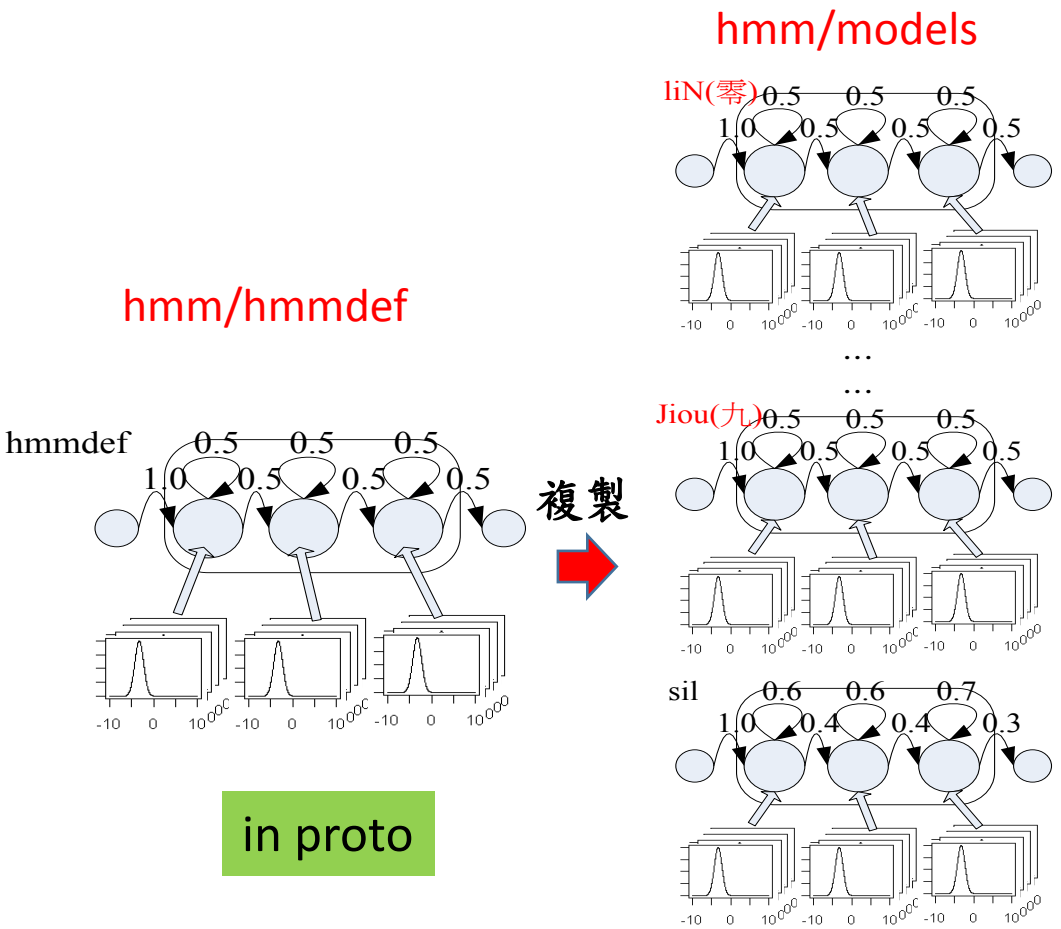
- bin/macro.c

  產生Variance floor macro (Vfloor)

  (為避免有訓練出之Gaussian variance的值underflow)，最後把所產生的Vfloor輸出為一個檔案到hmm資料夾 (檔名為：macro)。

- bin/models_1mixsil.c

執行 models_1mixsil.c 來建立**silence 的HMM model**，且**複製**0~9 的HMM model。最後把上述產生的所有HMM model輸出為一個檔案到hmm資料夾 (檔名為：models)。



HERest - Adjust HMMs

➢HERest -C lib/config.cfg -S scripts/training.scp -I labels/Clean08TR.mlf -H hmm/macros -H hmm/models -M hmm lib/models.lst (HTK BOOK 18.7)

- HERest指令用於HMM model的**訓練** (即HMM的Problem 3)
- 參數說明:
  - -C lib/config.cfg → 輸入之特徵資料的設定是存放在lib/config.cfg (與HCompV同)
  - -S scripts/training.scp → 輸入之training data的存放位置在：scripts/training.scp (與HCompV同)
  - -I labels/Clean08TR.mlf → Clean08TR.mlf內容為: training.scp內的每個training data的解答 (即對應的數字串)
  - -H hmm/macros → 從hmm資料夾, 讀取Vfloor檔案 macros
  - -H hmm/models → 從hmm資料夾, 讀取HMM模型的檔案 models, 來進行training (updating)
  - -M hmm → HERest完成training後, 會把輸出的HMM 檔案models 存放到 hmm 資料夾。(註: 如果不想把原有的HMM模型檔案models蓋掉，就改放到其他資料夾，不要放到hmm資料夾)
  - lib/models.lst → a list of word models (字彙檔, 不是字典檔)，,sil ,0,1,2 ….,9 (即在輸出之11個models內的HMM模型 其解答是屬於sil ,0,1,2 ….,9 的list內)*

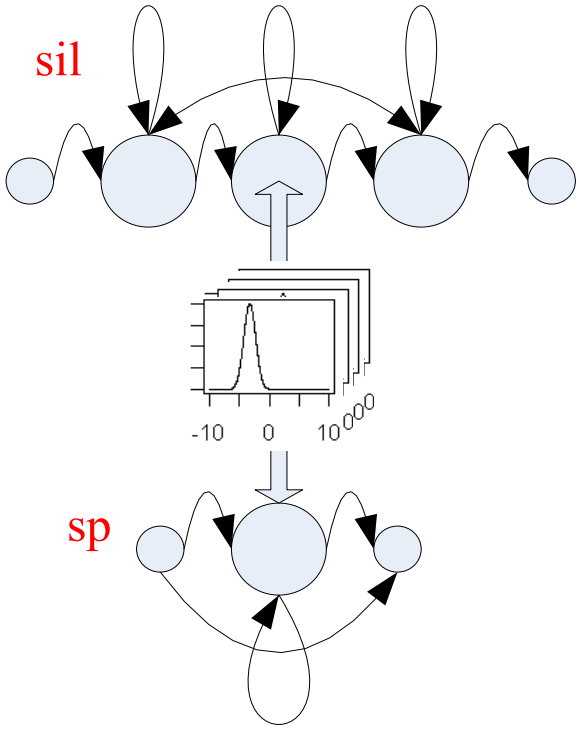
參考HTK book p164 (8.5)

Add SP Model

在開始與結束時會用到sil模型。
一個語音檔可能包含有許多個別數字的發音，這些不同數字發音之間會以sp來區別。

- bin/spmodel_gen.c

  執行spmodel_gen.c來建立Short Pause (SP) Model



執行方法如下：

spmodel_gen.c hmm/models hmm/models

→ 此C程式會以 hmm/models hmm/models 為參數。
→ 其中，第一個參數是**讀檔**(從hmm資料夾把models讀出)，而第二個參數是**寫檔**(將輸出檔models寫到hmm資料夾)。
→ 輸出檔的產生方式為：產生一個short pause HMM model，然後把它加到原有的models內(加入後, 在models內就有12個HMM 模型)。

HHEd - Modify HMMs

參考HTKBook 10.1

- HHEd -H hmm/macros -H hmm/models -M hmm lib/sil1.hed lib/**models_sp.lst**

  指令說明:更新修改後的HMM模型參數

lib/models_sp.lst → a list of word models (字彙檔)，
sp, sil ,0,1,2 ….,9 (詞彙資訊 {sil, sp, 0 ,1,..,9})

HERest - Adjust HMMs

- 新增short pause state 後，再利用HERest指令來進行HMM model training。
- HERest -C lib/config.cfg -S scripts/training.scp -I labels/**Clean08TR_sp**.mlf -H hmm/macros -H hmm/models -M hmm lib/**models_sp.lst**

比較第一次(尚未加short pause之前)，
→ HERest -C lib/config.cfg -S scripts/training.scp -I labels/**Clean08TR**.mlf -H hmm/macros -H hmm/models -M hmm lib/**models.lst**

★★

修改2：請增加高斯模型的數目以提高HMM的精確度。

Modification of Models

➢lib/mix2_10.hed

MU 2 {liN.state[2-4].mix}

MU 2 {#i.state[2-4].mix}

MU 2 {#er.state[2-4].mix}

MU 2 {san.state[2-4].mix}

MU 2 {sy.state[2-4].mix}

…

→ mix2_10.hed 存放在/lib資料夾

參考：HTK Book 18.8

MU: mixture up

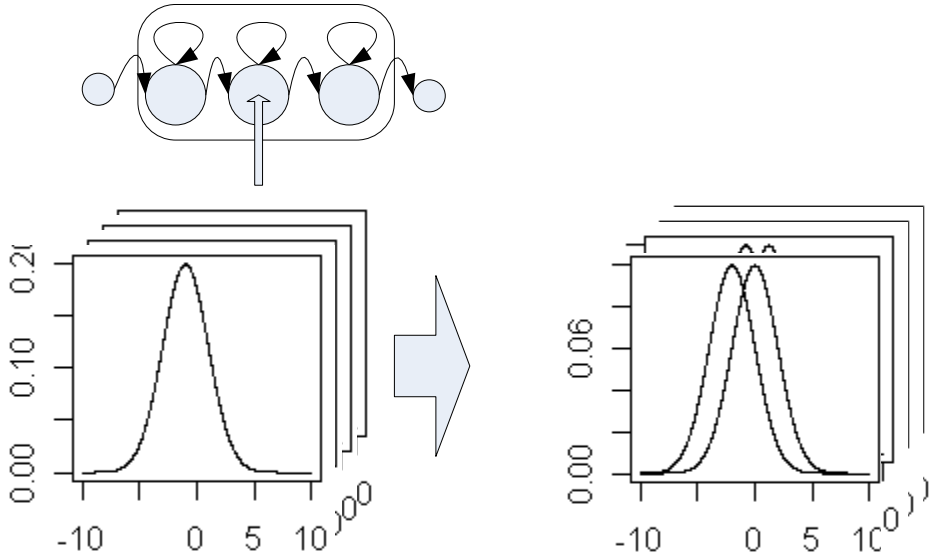
→ MU 2將數字0(liN)之GMM-HMM 模型的mixture component數目增加到**2**個（如果 MU **+2**，則是新增2個Gaussian pdf模型），state[2-4]指明：這件事僅針對state 2,3,4 (因為state 1,5是entry state與exit state)

在proto例子裏(p.11)，mixture component數為1，這裡增加Mixture Gaussian的數目來提高HMM的精確度。
(改變GMM數目時，proto不必更改。)

HHEd - Modify HMMs (Here, Number of Mixtures)

- HHEd -H hmm/macros -H hmm/models -M hmm lib/mix2_10.hed lib/models_sp.lst

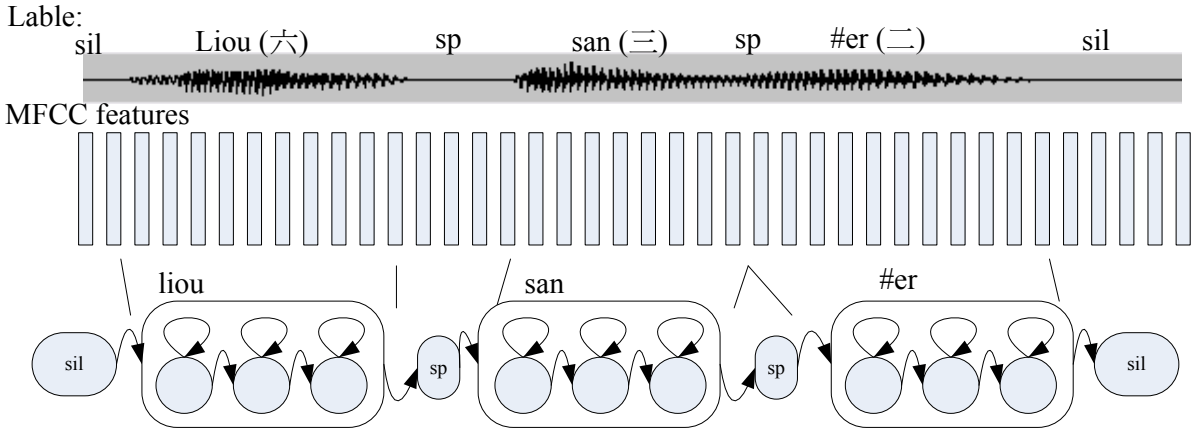
  指令說明:更新修改後的HMM模型參數



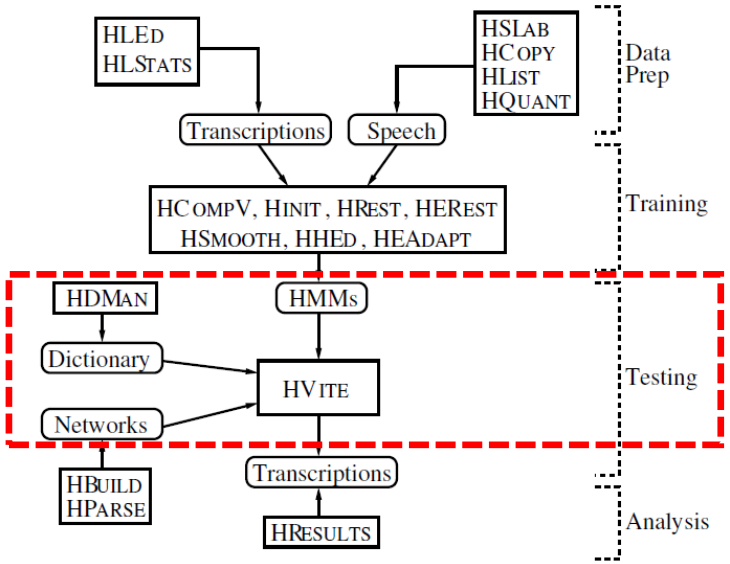
HERest - Adjust HMMs Again

- 新增Gaussian mixture component的數目後，再利用HERest指令來進行HMM model 的訓練。
- HERest -C lib/config.cfg -S scripts/training.scp -I labels/Clean08TR_sp.mlf

  -H hmm/macros -H hmm/models -M hmm lib/models_sp.lst


Lable:
sil
Liou (六)
sp
san (三)
sp
#er (二)
sil
MFCC features
liou
san
#er
sil
sp
sp
sil

Testing


HLED
HLSTATS
HSLAB
HCOPY
HLIST
HQUANT
Data Prep
Transcriptions
Speech
HCOMPV, HINIT, HREST, HEREST
HSMOOTH, HHED, HEADAPT
Training
HDMAN
HMMs
Dictionary
HVITE
Networks
Testing
HBUILD
HPARSE
Transcriptions
HRESULTS
Analysis

HParse - Construct Word Net

HTK Book 18.18

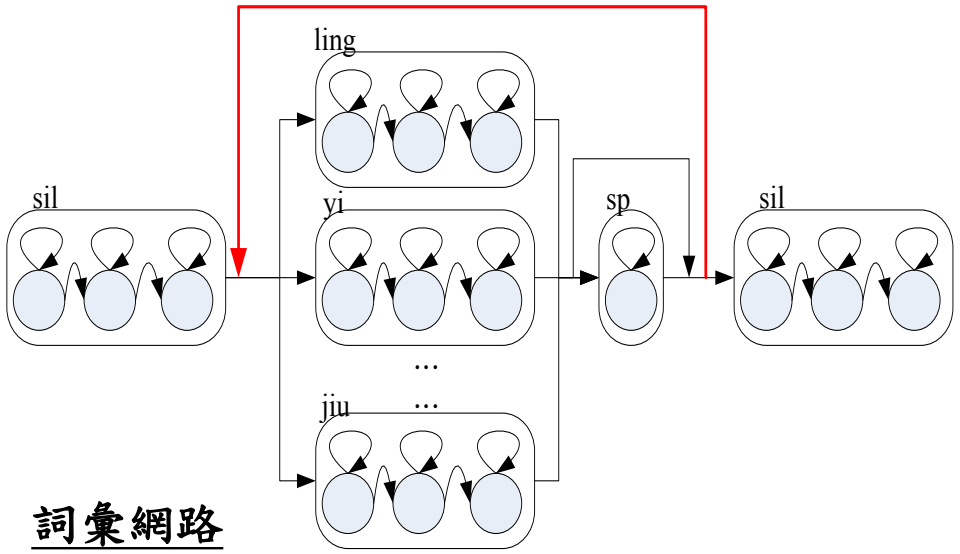
➢HParse lib/grammar_sp lib/wdnet_sp

指令說明:建立詞彙網路

參數說明:

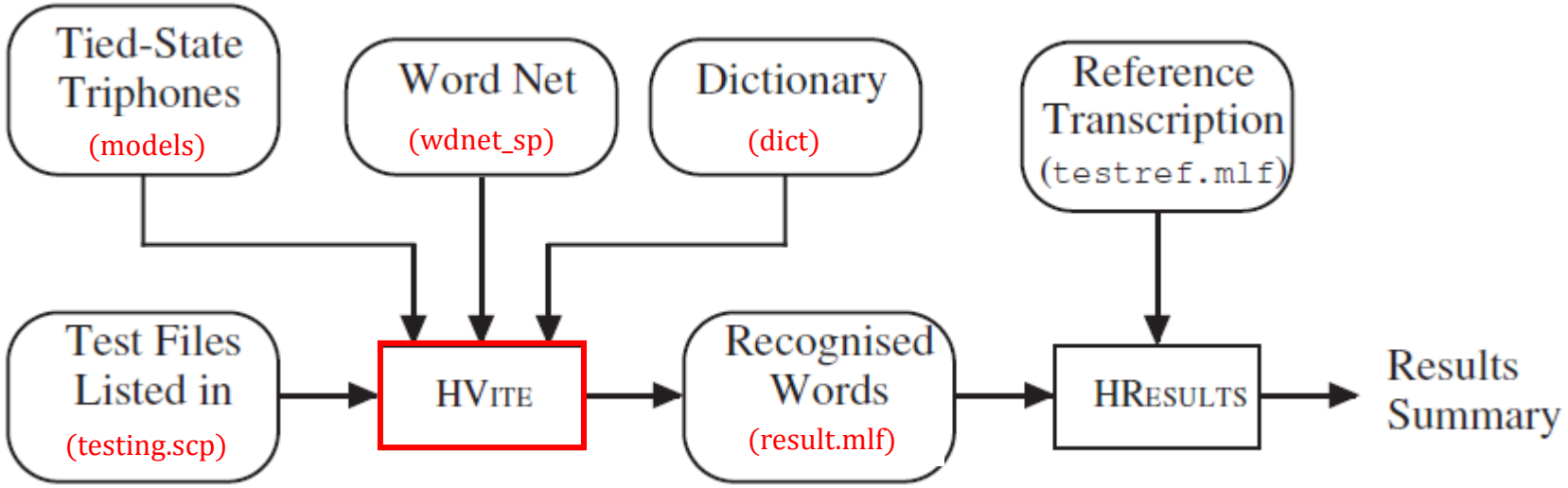
lib/grammar_sp : 數字辨識的 詞彙資訊 {sil, sp, 0 ,1,..,9}

lib/wdnet_sp: 產生之 詞彙網路 的名稱



詞彙網路

Testing


Tied-State Triphones
(models)
Word Net
(wdnet_sp)
Dictionary
(dict)
Reference Transcription
(testref.mlf)
Test Files Listed in
(testing.scp)
HVITE
Recognised Words
(result.mlf)
HRESULTS
Results Summary

HVite - Viterbi Search

HTK Book 18.25

➢ HVite -H hmm/macros -H hmm/models -S scripts/testing.scp -C lib/config.cfg
-w lib/wdnet_sp -l '*' -i result/result.mlf -p 0.0 -s 0.0 lib/dict lib/models_sp.lst

指令說明: 將testing data 使用Viterbi進行辨識

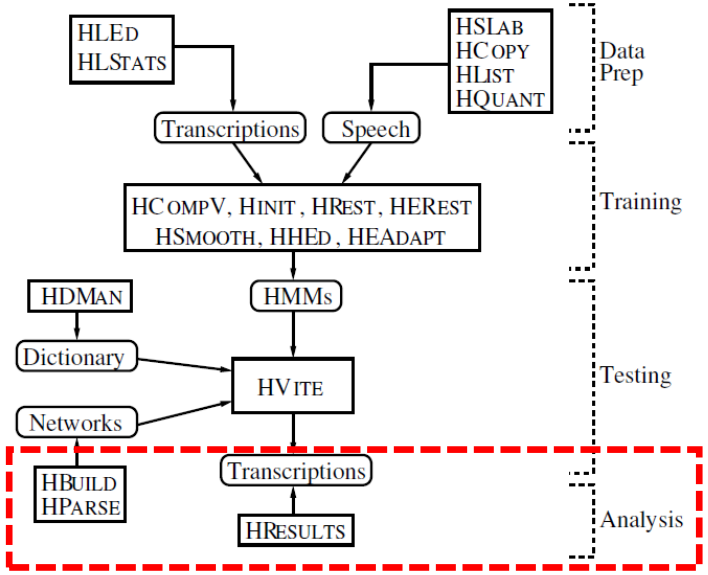
參數說明:

- -w lib/wdnet_sp (input word net)
- -i result/result.mlf (將辨識的結果，輸出至result/result.mlf)
- lib/dict (字典檔，數字對應phone)
- lib/models_sp.lst (字彙檔：sil , sp ,0, 1, …, 9 的labels)
- -p and -s set the word insertion penalty and the grammar scale factor
- -l '*' will cause a label file named xxx to be prefixed by the pattern "*/xxx" in the output MLF file

"*/N720511.lab" (答案的label) → "*/N720511.rec" (Viterbi result label)

↳ 請參閱 labels\answer.mlf 的內容

Analysis


HLED
HLSTATS
HSLAB
HCOPY
HLIST
HQUANT
Data Prep
Transcriptions
Speech
HCOMPV, HINIT, HREST, HEREST
HSMOOTH, HHED, HEADAPT
Training
HDMAN
HMMs
Dictionary
HVITE
Testing
Networks
HBUILD
HPARSE
Transcriptions
HRESULTS
Analysis

HResults - Compared With Answer

HTK Book 18.21

➢HResults -e “???” sil -e “???” sp -I labels/answer.mlf lib/models_sp.lst result/result.mlf

指令說明:將Viterbi判斷的結果與答案進行比對(求正確率、準確率)

參數說明:

- -I labels/answer.mlf (answer.mlf的內容為各.wav聲音檔的解答)
- lib/models_sp.lst (字彙檔：sil , sp ,0, 1, …, 9 的labels)
- result/result.mlf (Viterbi 所求出的結果)
- -e “???” sil -e “???” sp (用以忽略sil以及sp的label ，因計算識別率(分析)時，sil、sp是不需考慮的)

HTK Book p232

HResults

HTK Book 3.4.1

- 執行結果:

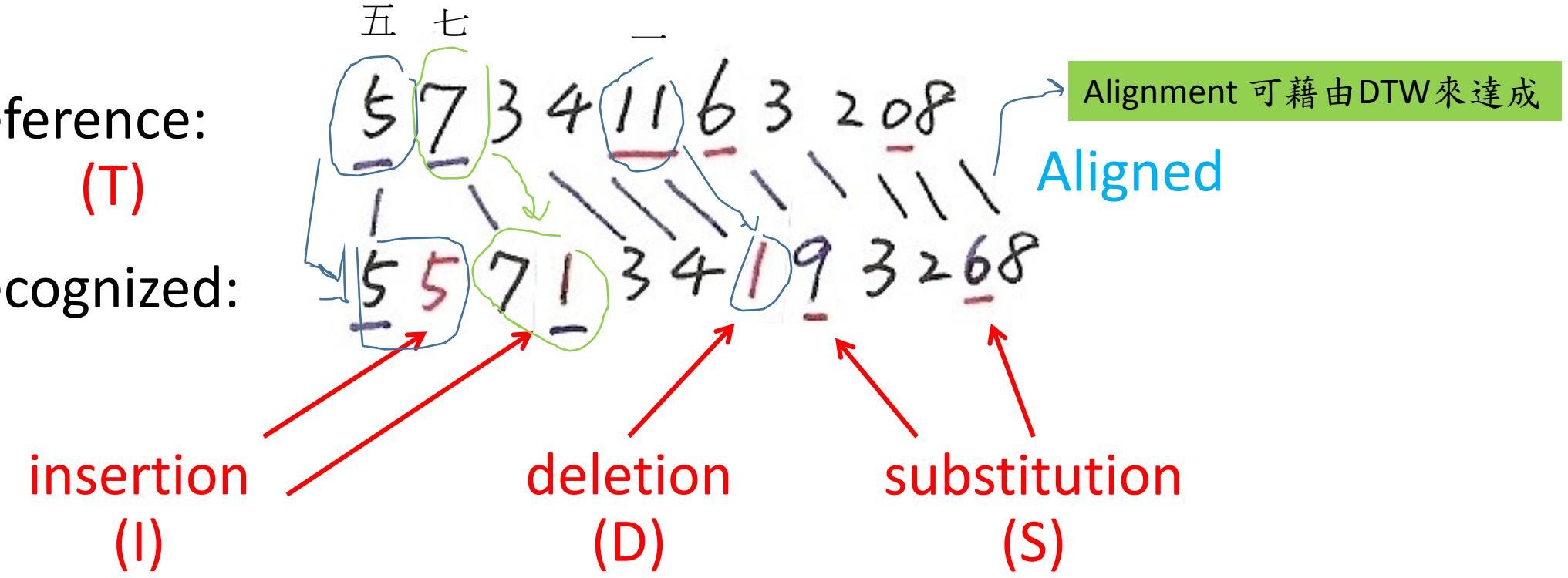
```
====================== HTK Results Analysis =======================
  Date: Wed Apr 17 00:26:54 2013
  Ref : labels/answer.mlf
  Rec : result/result.mlf
---------------------- Overall Results ----------------------
SENT: %Correct=38.54 [H=185, S=295, N=480]
WORD: %Corr=96.6[illegible], Acc=74.34 [H =1679, D=13, S=46, I=387, N=1738]
===================================================================
```

S:Substitution errors ，**D**:deletion errors，**I**:insertion errors ， **N**:Total Sample ，**H**:correctly recognized

%Corr: percentage correct = $$\text{Percent Correct} = \frac{N - D - S}{N} \times 100\%$$

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

Recognition Errors



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

Procedure for Experiment

作業請上傳： cat accuracy 所查看到的辨識結果與以下三個檔案

- 03_training.sh
- lib/mix2_10.hed
- lib/proto

方案一 使用virtualBox
(配合 virtualBox安裝教學.ppt 說明檔)

建議使用此方案

實驗步驟 (1/3)

已將**本地端**建立的資料夾folder與虛擬機下建立的資料夾**share** 連接在一起。

- 將dsp_hw2解壓縮至本地端電腦的共用資料夾folder裡
- 打開virtualBox ，切換到 /share/dsp_hw2 目錄

  cd /share/dsp_hw2 → cd指令後要空一格
- 並輸入 ls 查看dsp_hw2資料夾內容

```
kaldi@kaldi:~/share/dsp_hw2$ ls
00_clean_all.sh   03_training.sh  clean_model.sh  lib      scripts          TraFile.c
01_run_HCopy.sh   04_testing.sh   hmm             MFCC     set_htk_path.sh
02_run_HCompV.sh  bin             labels          result   speechdata
```

Shell腳本解釋器有bash、sh、csh、ksh等

實驗步驟 (2/3)

- 步驟1→ bash 01_run_HCopy.sh → 1. Data Prep

```
kaldi@kaldi:~/share/dsp_hw2$ bash 01_run_HCopy.sh _
```

- 步驟2→ bash 02_run_HCompV.sh
- 步驟3→ bash 03_training.sh

→ 2., 3. Training

- 步驟4→ bash 04_testing.sh

→ 4. Testing & Analysis

實驗步驟 (3/3)

- cd result
- 輸入cat accuracy 可查看辨識結果 (辨識率74.34)

```
user@user-PC /home/dsp_hw2/result
$ cat accuracy
====================== HTK Results Analysis =======================
  Date: Sat May  6 17:08:49 2017
  Ref : labels/answer.mlf
  Rec : result/result.mlf
------------------------ Overall Results --------------------------
SENT: %Correct=38.54 [H=185, S=295, N=480]
WORD: %Corr=96.61, Acc=74.34 [H=1679, D=13, S=46, I=387, N=1738]
===================================================================
```

方案二 使用Cygwin
(配合 [1]Cygwin安裝教學.ppt 、[2]在Cygwin 底下上安裝HTK教學.ppt)

實驗步驟 (1/3)

Cygwin不需進行**本地端**資料夾與虛擬機下資料夾的連接。

- 將dsp_hw2解壓縮至C:\cygwin64\tmp
- 打開cygwin ，切換到 /tmp/dsp_hw2 目錄

  cd /tmp/dsp_hw2
- 並輸入 ls 查看dsp_hw2資料夾內容

cygwin64為cygwin的根目錄

cd指令後要空一格

```
user@user-PC /tmp/dsp_hw2
$ ls
00_clean_all.sh  02_run_HCompV.sh  04_testing.sh  bin  labels  result   set_htk_path.sh
01_run_HCopy.sh  03_training.sh    MFCC           hmm  lib     scripts  speechdata
```

Shell腳本解釋器有bash、sh、csh、ksh等

實驗步驟 (2/3)

- 步驟1→ sh 01_run_HCopy.sh

```
user@user-PC /tmp/dsp_hw2
$ sh 01_run_HCopy.sh
```

→ 1. Data Prep

- 步驟2→ sh 02_run_HCompV.sh
- 步驟3→ sh 03_training.sh

→ 2., 3. Training

- 步驟4→ sh 04_testing.sh

→ 4. Testing & Analysis

實驗步驟 (3/3)

- cd result
- 輸入cat accuracy 可查看辨識結果 (辨識率74.34)

```
user@user-PC /home/dsp_hw2/result
$ cat accuracy
====================== HTK Results Analysis =======================
  Date: Sat May  6 17:08:49 2017
  Ref : labels/answer.mlf
  Rec : result/result.mlf
------------------------ Overall Results --------------------------
SENT: %Correct=38.54 [H=185, S=295, N=480]
WORD: %Corr=96.61, Acc=74.34 [H=1679, D=13, S=46, I=387, N=1738]
===================================================================
```

練習

- 藉由調整“HMM**模型參數**”和“**訓練的iteration**”來提高辨識率

- 方法:
  1. 調整Prototype(簡報**第11頁的修改1**)
  2. lib/mix2_10.hed (簡報**第17頁的修改2**)
  3. 增加training iteration (如**下頁**說明)

```
====================== HTK Results Analysis =======================
 Date: Sat May  6 21:00:57 2017
 Ref : labels/answer.mlf
 Rec : result/result.mlf
------------------------ Overall Results --------------------------
ENT: %Correct=71.25 [H=342, S=138, N=480]
ORD: %Corr=96.43, Acc=90.22 [H=1676, D=20, S=42, I=108, N=1738]
===================================================================
```

方法3: Add Training iteration

修改 i 迴圈的終止條件 (即第9頁中的N值，整個完整的training data使用N次）

- 開啟 03_training.sh

```
################################################################
# re-adjust mean, var
echo "step 01 [HErest]: adjust mean, var..."
for ((i=1;i<=3;i++))
do
        echo "iteration $i"
        HERest -C $config -I $label \
                -t 250.0 150.0 1000.0 -s $data_list \
                -H $macro -H $model -M $mmf_dir $model_list
done
```

for ((i=1;i<=3;i++)) 代表HERest會執行3次，即3次iteration。

在o3_training.sh中，i 迴圈的終止條件 (iteration次數) 的調整，有以下三部分：
re-adjust mean, var
add short pause model, change model_list and label file
increase mixture

註

- 調整HMM Prototype後（指p.11，改變state數目或改變transition probability matrix），必須刪除舊的model（models與macros），而這動作可在cygwin command line 輸入 sh clean_model.sh 來執行。移除後，再重複 步驟2~4 進行HMM初始化、訓練、測試等動作(步驟1是擷取特徵所以不用再次執行)
- **至少要達到90%的正確率，才算完成作業。**

Reference

- Assignment 2-1 HMM Training and Testing of Digital Speech Processing, Lin-shan Lee