

Digital Speech Processing

HW-2-1

歐政鷹 資工所碩一 R07922142

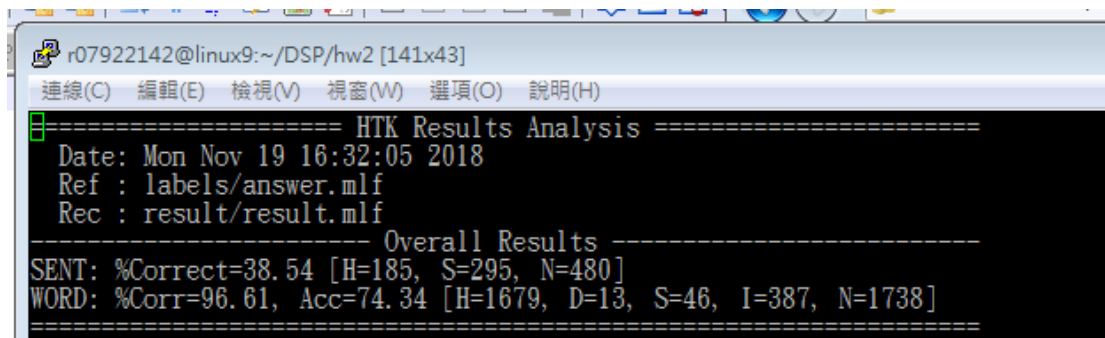
執行環境：

程式以 C++ 語言為基礎，並以 CSIE workstation 作為執行環境，程式基本上都是在 linux9 機器上執行，對於 linux9 的環境配置及 g++ 版本可參考下圖。

```
r07922142@linux9 {~/DSP/hw1} [W0] whoami
r07922142
r07922142@linux9 {~/DSP/hw1} [W0] uname -a
Linux linux9 4.18.12-arch1-1-ARCH #1 SMP PREEMPT Thu Oct 4 01:01:27 UTC 2018
x86_64 GNU/Linux
r07922142@linux9 {~/DSP/hw1} [W0] g++ -v
使用內建 specs。
COLLECT_GCC=g++
COLLECT_LTO_WRAPPER=/usr/lib/gcc/x86_64-pc-linux-gnu/8.2.1/lto-wrapper
目的：x86_64-pc-linux-gnu
配置為：/build/gcc/src/gcc/configure --prefix=/usr --libdir=/usr/lib --libex
ecdir=/usr/lib --mandir=/usr/share/man --infodir=/usr/share/info --with-bugu
rl=https://bugs.archlinux.org/ --enable-languages=c,c++,ada,fortran,go,lto,o
bjc,obj-c++ --enable-shared --enable-threads=posix --enable-libmpx --with-sy
stem-zlib --with-isl --enable-_cxa_atexit --disable-libunwind-exceptions --
enable-clocale=gnu --disable-libstdc++-pch --disable-libssp --enable-gnu-uni
que-object --enable-linker-build-id --enable-lto --enable-plugin --enable-in
stall-libiberty --with-linker-hash-style=gnu --enable-gnu-indirect-function
--enable-multilib --disable-werror --enable-checking=release --enable-defaul
t-pie --enable-default-ssp --enable-cet=auto
執行緒模型：posix
gcc version 8.2.1 20180831 (GCC)
```

linux9 環境配置及 g++ 版本

Baseline：



```
r07922142@linux9:~/DSP/hw2 [141x43]
連線(C) 編輯(E) 檢視(V) 視窗(W) 選項(O) 說明(H)
===== HTK Results Analysis =====
Date: Mon Nov 19 16:32:05 2018
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]
=====
```

Baseline 執行結果

構想與測試：

我初期的想法是透過調整 Iteration > state > Gaussian 來改善準確率，一開始我打算先把 3 個 Iteration 均以相同的次數來進行測試，我以 5 的倍數測試了 5~30 次 iteration 對於準確率的影響，測試結果如下表。

Iteration	5	10	15	20	25	30
Accuracy(%)	75.83	77.79	77.96	77.16	77.68	77.73

Iteration 次數比較表

從上表的測試結果中，可以看見在 iteration 次數為 15 的時候，可以有較高的準確率，但實際上在 iteration 次數為 10,15,20,25,30 時，準確率都是落在 77~78%之間，所以我認為 3 個 Iteration 均以相同的次數來訓練只能把準確率提升 3%左右。

之後，我打算根據一開始的設定，以 1:1:2 的方式來測試 iteration 次數對準確率的影響，我分別以(6,6,12),(10,10,20),(15,15,30),(20,20,40)的 iteration 次數進行了測試，測試結果如下表。

Iteration	6,6,12	10,10,20	15,15,30	20,20,40
Accuracy(%)	77.16	77.96	76.41	76.35

Iteration 次數比較表

從上表的測試結果中，可以看見在 iteration 次數為 10 時候，可以有較高的準確率，但跟前一個測試比起來，準確率是沒有提升的，兩種測試的最高準確率都落在 77.96%，所以我認為 iteration 次數的改變不能有效的提升準確率，最多只能提升 3%左右。

我認為對 iteration 次數的測試已經足夠。再下一步，我開始測試 state 數目對準確率的影響，為了加快測試速度，這邊我是以預設的(3,3,6) 的 iteration 次數，並以 5 的倍數測試了 5~20 個 state 時的情況，測試結果如下表。

Iteration	5	10	15	20
Accuracy(%)	74.34	93.67	95.91	94.42

state 數目比較表

從上表的測試結果中，可以發現更改 state 數目可以令到準確率有顯著的提升，state 數目從 5 提升到 10 時就提高了約 20%的準確度，在 state 數目為 15 時

準確率就超過了 95%。由此，我測試了 state 數目為 15~20 時的準確率，希望找出最高準確率的位置。

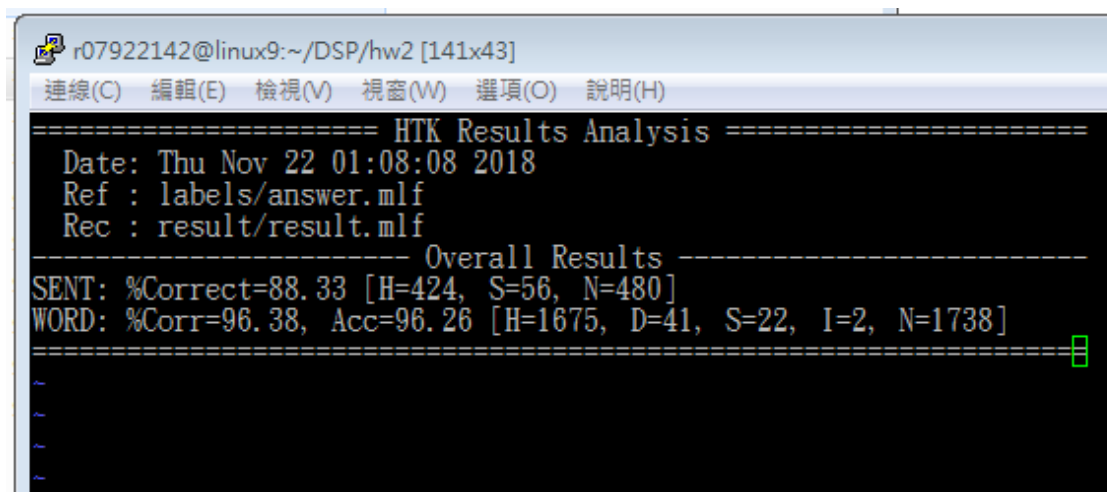
Iteration	15	16	17	18	19	20
Accuracy(%)	95.91	95.97	95.57	95.11	95.05	94.42

state 數目比較表

經過測試後，找出了 state 數目為 16 時可以得到最好的準確率，所以，我決定以 state 數目為 16，iteration 次數為(10,10,20) 作測試，看看準確率會不會再提高。

執行結果：

最終以 state 數目為 16，iteration 次數為(10,10,20) 作測試的結果，準確率提高到 96.26%，我決定把這個結果作為我的最終結果。

A screenshot of a terminal window titled 'r07922142@linux9:~/DSP/hw2 [141x43]'. The window has a menu bar with '連線(C)', '編輯(E)', '檢視(V)', '視窗(W)', '選項(O)', and '說明(H)'. The terminal output shows 'HTK Results Analysis' with the following details: Date: Thu Nov 22 01:08:08 2018, Ref: labels/answer.mlf, Rec: result/result.mlf. Under 'Overall Results', it shows SENT: %Correct=88.33 [H=424, S=56, N=480] and WORD: %Corr=96.38, Acc=96.26 [H=1675, D=41, S=22, I=2, N=1738]. The terminal has a green cursor at the end of the last line.

```
r07922142@linux9:~/DSP/hw2 [141x43]
==== HTK Results Analysis ====
Date: Thu Nov 22 01:08:08 2018
Ref : labels/answer.mlf
Rec : result/result.mlf
----- Overall Results -----
SENT: %Correct=88.33 [H=424, S=56, N=480]
WORD: %Corr=96.38, Acc=96.26 [H=1675, D=41, S=22, I=2, N=1738]
```

最終執行結果

最終參數：

- States 數量：16
- Gaussian mixtures：預設
- Iteration 次數：(10 , 10 , 20)

辨識率：

- 96.26%