r07922106 曾俊為 DSP_HW3 Report

這次作業原本我打算直接在我的 laptop 上建置 SRILM 的環境來處理，一開始遇到一直找不到 srilm-1.5.10 路徑的問題，後來才發現原來他是要吃絕對路徑，不過再寫好 mapping.py 之後，要跑 disambig 時又發現一直沒辦法 link 到 SRILM 的 library，所以後來索性直接拉 docker 的 image 下來寫作業。而由於不熟悉 docker 環境以及各種指令的用法，所以也花了些時間摸索。
以下為環境及編譯執行流程:

Environment:

Docker

```
docker run -it /Desktop/數位語音處理/dsp_hw3:/root/dsp_hw3 ntudsp2020autumn/sirim
```

Compile:

make

```
root@8f42a0794c3c:~/dsp_hw3/dsp_hw3# make
```

Execution:

1. ZhuYin-Big5.map

```
dsp_hw3# make map FROM=Big5-ZhuYin.map TO= ZhuYin-Big5.map
```

2. Run mydisambig

```
./mydisambig test_data/1_seg.txt ZhuYin-Big5.map bigram.lm  1_result.txt
```

3. Check result

```
dsp_hw3# vim 1_result.txt
```

Mapping.py 一開始做的就是將原始 Big5-ZhuYin.map 資料的中文和注音分開，有破音字的需要特別處理，接下來再由注音頭去做分類對應，最後輸出成 ZhuYin-Big5.map。
比較需要注意的就只有 encoding 的時候要設為 big5-hkscs，其他就是簡單的引用參數及讀寫檔。
而 mydisambig 的部分，先去處理已經做過 segmentation 的 test data，再將 map 以及 language model 讀進來後，就可以用 Viterbi algorithm 來找出機率最高的 sequences，接下來再用 map 的 API 來連結注音文和預測的中文字，最後利用 vectors 的結構將 Viterbi algorithm 所算出的最佳解呈現出來。
另外，output 需要注意每行開頭為<s>、結尾為</s>以及每個字中間都有空格即可。