DSP HW3 Report

R09944072 鍾宜樺

- Compile and Execute (Makefile)

  make ====> compile mydisambig.cpp to exe file

  make map ====> use Big5-ZhuYin.map to generate ZhuYin-Big5.map

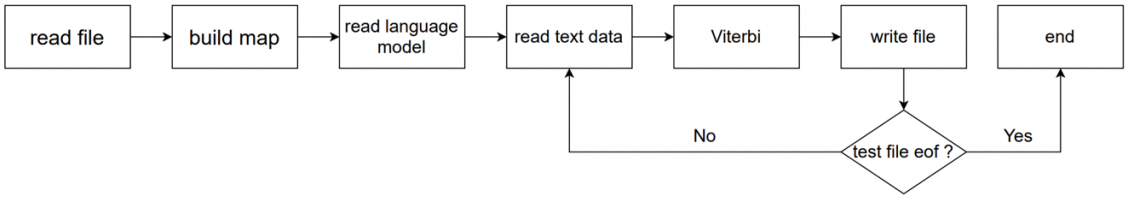
  make run ====> exe mydisambig to run all test_data (segmented)

- Implementation

mapping.py 的部分就是很直觀的用 dictionary 實作，每讀一行就分 key, value，最後再寫入第二個檔案裡面。

mydisambig.cpp

簡單畫一下 mydisambig.cpp flowchart:



這部分助教在 spec 上有推薦使用一些 SRILM 的資料結構可用，我使用了 SRILM 提供的 Ngram library。ZhuYin-Big5 的 map 則是用 C++ map<string, vector<string>> m 做。在 Viterbi 的部分，開了兩個型別不同的二維 vectors: vector<vector<double>> v, vector<vector<int>> bt：分別紀錄 Viterbi 的整個 map 和 best path，因為用 vector 寫不用額外處理字串長度、寫入東西時 .push_back 也比較方便。Viterbi recursion 的部分將計算 bigram probability 的部分另外寫成 get_bigram_prob function，每次回傳 map 上面兩個 words 之間走過去的機率，一但找到更好的路徑則取代當前的最佳路徑。最後利用 recursion 中紀錄的 bt vector 做 back trace 的動作完成 Viterbi。

最後將 disambig vs. MyDisambig 跑出來的結果分別存在不同的 file (resilt1, result2) 下，寫個 bash 去 diff 所有檔案確認程式運行的正確性。

```
for i in {1..10..1}
do
        echo "$i"
        #perl separator_big5.pl ./test_data/"$i".txt > ./test_data/"$i"_seg.txt
        #disambig -text test_data/"$i"_seg.txt -map ZhuYin-Big5.map -lm  bigram.lm  -order 2 > result1/"$i".txt
        diff result1/"$i".txt result2/"$i".txt
done
```

- 作業中遇到的困難

有鑒於上次作業二的部分無法在 M1 上面寫作業，這次的作業一樣是跑在 AWS 提供的免費 server 上。Docker 直接 pull image 也省去很多裝環境的時間。後來聽朋友說，CSIE server 上面好像可以跑 SRILM！

這份作業花比較多時間在搞懂作業整個流程，一開始很順利的 pull image 想從 Makefile 直接看懂也覺得很不容易。雖然有個大致的概念，但實際上還沒有特別了解該怎麼做。最後拿著自己畫的流程圖去請教之前已經修過的學長姐 mydisambig，確定每個部分的實作流程後再開始 implement。

另外是直接複製助教的 spec *ngram-count* 的第一行 command 時，好像會複製到看不到的特殊字元，所以 command 執行之後也不會有 lm.cnt 作為 output，但 command 不會噴 error，一開始很疑惑，解決方法只是換用手打。

最後是其實我一開始想用 SRILM 提供的 libraray 可以用到的資料結構完成這次作業，但最後因為 index 的部分、像是 array size 相關的地方做不好，就把整份 code 打掉重練了。

- **比較 disambig vs. MyDisambig**

最直觀的差別是 disambig 運行的速度比我的 MyDisambig 快，效能上還是有蠻大的差距，尤其是到後面的測資會比較嚴重。像是第 10 筆測資

```
real    0m7.801s
user    0m7.738s
sys     0m0.024s
```

disambig

```
real    0m11.952s
user    0m11.855s
sys     0m0.048s
```

MyDisambig

表格因為版面限制兩頁，所以總共放前四筆測資)：

test_data	1	2	3	4
disambig	real 0m2.833s user 0m2.752s sys 0m0.073s	real 0m5.250s user 0m5.194s sys 0m0.041s	real 0m3.466s user 0m3.423s sys 0m0.028s	real 0m4.341s user 0m4.298s sys 0m0.028s
MyDisambig	real 0m3.794s user 0m3.712s sys 0m0.065s	real 0m7.514s user 0m7.437s sys 0m0.057s	real 0m5.094s user 0m5.022s sys 0m0.057s	real 0m5.955s user 0m5.882s sys 0m0.058s

另外還有一個我覺得很有趣的地方是，一開始想要看自己跑出來的 result2/1.txt 有沒有對，就點開來看發現第一個詞：「華視」會被轉成「忽視」，一開始在想可能是自己哪裡寫錯了，但後來跑了 SRILM 的 disambig 發現竟然也會轉成忽視。我覺得是因為 corpus.txt 字彙量不足，導致在走 Viterbi 的時候會認為「忽視」的機率比較高。

Ref: SRILM Manual Pages