DSP HW3

B02902015 資工三 梁智泓

(1) What is your environment?

系上的工作站 Linux 64bit

(2) How to "make" your program?

有兩個檔案需要 compile, 分別是 mapping.cpp 與 mydisambig.cpp

1. 若要 compile mapping.cpp:
   在檔案路徑下輸入: **make compile_map**
2. 若要 compile mydisambig.cpp:
   在檔案路徑下輸入: **make**

(3) How to "execute" your program?

1. 初始化: **make clean**
2. 將 Big5-ZhuYin.map → ZhuYin-Big5.map:
   在檔案路徑下輸入: **make map**
3. 透過 mydisambig 跑 10 個測試檔
   （注意! 10 個測試檔必須放在名為 testdata 的資料夾底下，且先透過 separator_big5.pl 切好，分別命名為$(i).txt, i=[1, 10]）
   做完以上前置作業後，便可以直接在檔案路徑下輸入: **make run**
   就會直接將生成的結果放在 result1 的資料夾裡，且各檔案名字為$(i).txt, i=[1, 10]
4. 若要將要手動輸入執行 mydisambig:
   只要在檔案路徑下輸入: ./mydisambig –text $(file) –map $(map) –lm $(LM) –order $(order) > $(output_file)
   Ex: ./mydisambig –text ./input.txt –map ZhuYin-Big5.map –lm bigram.lm –order 2 > output.txt
5. 若要透過 SRILM 的 disambig 生成標準答案，只要在檔案路徑下輸入: **make ans**
   就會直接將生成結果放在 result2 的資料夾裡，且各檔案的名字為$(i).txt, i=[1, 10]
6. 若要進行比對，只要在檔案路徑下輸入: **make check**
   他便會將 result1/ 與 result2/ 的檔案比對，若完全相同，則只會有 Checking $(i).txt 顯示，不會出現其他文字

(4) What have you done?

- Big5-ZhuYin.map → ZhuYin-Big5.map: 將"注音→字"或"字→字"的對應存在 map 中，存之前確認是否有曾經存過，若有存過，就表示這組對應有出現過，所以將其濾掉，最後再將其輸出
- mydisambig: 將 input_text 一行行讀，在每行的頭跟尾分別加入<s>與</s>，透過 ZhuYin-Big5.map 找到句中每個注音或字可能對到的 candidate，再兩兩做 bigram 的 Viterbit，找出最大的可能路徑，並且將其輸出。
- 加速: 因為如果將 map 中所有可能的 candidate 都跑過一次會太慢，所以我做了部分的優

化，對有多個 candidate 的字，將 candidate 中沒出現在 voc 字庫裡的給去掉，這樣讓原本 13000 多不同的字只剩下 5000 多，大幅的加速了整體的運算時間。