

一、環境：

本次作業是在 CSIE workstation 完成，使用 linux9.csie.ntu.edu.tw 工作站（gcc 版本為 8.2.1）

```
gcc -A6420U 8'3'0 (gcc)
歡迎使用：bo2tx
-6u9p7e-c6f=9nro
9JT-JTPR6L7L --MIGU-JTUK6L-192U-27J7e=8un --6u9p7e-8un-tuq7ecf-1nucc7ou --6u9p7e-wn7c77p --qt29p7e-m6LLOL --6u9p7e-c76ck7u8=676926 --6u9p7e-q619n7f-b76 --6u9p7e-q619n7f-22b --
-JTPR6L7L-6xc6b7ou2 --6u9p7e-c7oc97e=8un --qt29p7e-JTP27c7cx-bcu --qt29p7e-JTP22b --6u9p7e-8un-nuidn6-op7ecf --6u9p7e-JTUK6L-pn77q-7q --6u9p7e-7ro --6u9p7e-b7n87u --6u9p7e-tu27
ol8\ --6u9p7e-79u8n98e2=c'c++'9q9'1o7c7su'8o'7ro'op7c'op7-c++ --6u9p7e-27u97e6 --6u9p7e-7u7e9q2=bo2tx --6u9p7e-JTP7ubx --MIGU-27276w-77p --MIGU-727 --6u9p7e-cx9'976x7f --qt29p7e
圖畫：\pn77q78cc72c78cc7cou778n76 --b7e77x=7n27 --JTP77L=7n2777p --JTP76x77L=7n2777p --7u7o77L=7n27727u97e7u7o --MIGU-pn87n7J=77f7b2:\7pn82'97c777u7x'
目錄：x8e'eq-bc-JTunx-8un
CORRECT77LO'7u7ybbE8=\n2777p78cc7xge'eq-bc-JTunx-8un78'3'0'77ro-7u7b7e7
CORRECT77CC=8++
退出目錄 2b7e2+
401235008@JTunx8 [-7q2b77u73] 8++ -A
```

二、How to “compile” your program

將 Makefile 中的 SRIPATH 與 MACHINE_TYPE 設定好，使用 make 指令（即「輸入 make 於終端介面」）即可，或也可以在設定好環境變數的情況下使用以下指令執行：

```
g++ -O3 -I$(SRIPATH) -w --std=c++11 -c mydisambig.cpp
g++ -o mydisambig mydisambig.o $(SRIPATH)/lib/$(MACHINE_TYPE)/liboolm.a
$(SRIPATH)/lib/i$(MACHINE_TYPE)/libdstruct.a /$(SRIPATH)/lib/i$(MACHINE_TYPE)/libmisc.a
```

三、How to “execute” your program

1. make map: 執行 mapping.py，產生 ZhuYin-Big5.map
python mapping.py Big5-ZhuYin.map ZhuYin-Big5.map
2. make separate: 使用作業檔案提供的 separator_big5 製作 corpus_seg.txt
perl separator_big5.pl corpus.txt > corpus_seg.txt
3. make bigram: 使用 srilm 提供的 ngram-count 對已分割過的檔案製造出 bigram.lm
\$(SRIPATH)/bin/\$(MACHINE_TYPE)/ngram-count -text corpus_seg.txt -write lm.cnt -order 2
\$(SRIPATH)/bin/\$(MACHINE_TYPE)/ngram-count -read lm.cnt -lm bigram.lm -unk -order 2

如果沒有對 txt 進行分割請先執行 mv testdata testdata_old、perl separator_big5.pl
testdata_old/\${i}.txt > testdata/\${i}.txt

4. make test: 使用 srilm 提供的 disambig 解析出答案，並存於\$(TESTDIR)，預設為 result1
[-d \$(TESTDIR)] || mkdir -p \$(TESTDIR);
@for i in \$(shell seq 1 10); do \
\$(SRIPATH)/bin/\$(MACHINE_TYPE)/disambig -text \$(TESTDATA)/\${i}.txt -map \$(TO) -lm
\$(LM) -order 2 > \$(TESTDIR)/\${i}.txt; \
done;
5. make run: 執行 mydisambig 並把結果存於\$(MYDISAMBIGDIR)，預設為 result2
[-d \$(MYDISAMBIGDIR)] || mkdir -p \$(MYDISAMBIGDIR);
for i in \$(shell seq 1 10); do \
echo "Running \${i}.txt"; \
./mydisambig -text testdata/\${i}.txt -map \$(TO) -lm \$(LM) -order 2 >
\$(MYDISAMBIGDIR)/\${i}.txt; \
done;
6. make clean: 清除之前 make 產生的檔案
rm -f mydisambig.o mydisambig

二、What you have done

我按照課程網站的 Q&A 修正了 `makefile`，並且以 SRILM 將注音轉為國字
並在 `mydisambig` 中實作了 `viterbi` 演算法，並對其結果與 SRILM 的結果比對