

# Digital Speech Processing Homework 3

May 09 2018

黃淞楓



# To complete this homework, you need to...

- Build a character-based language model with toolkit SRILM.
- Decode the ZhuYin-mixed sequence.

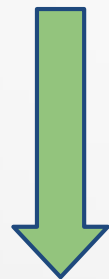


# Outline

- Introduction
- Flowchart
- SRILM
- Step by Step
- Submission and Grading

# Introduction

讓他十分ㄟ怕  
只ㄟ望ㄟ己明ㄟ度別再這ㄟㄟ命了  
演一ㄟ樂產一ㄟ人積ㄟㄟ型提ㄟ競爭ㄟ

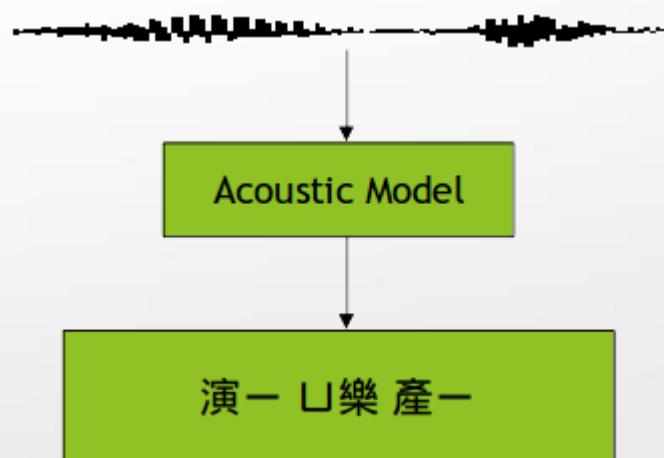
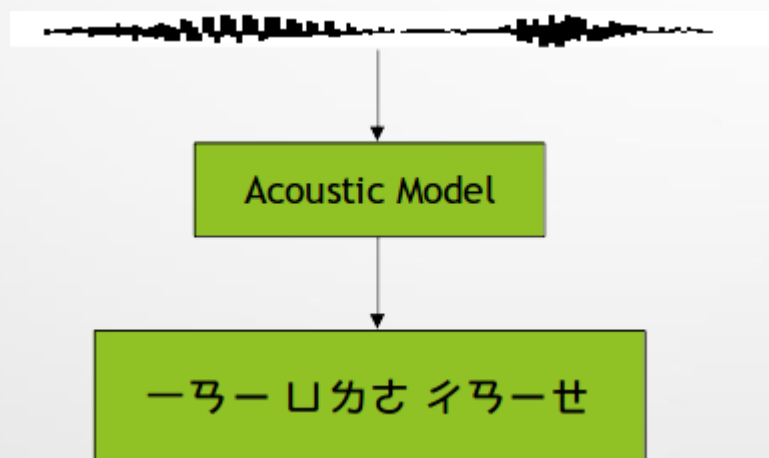


HW3：注音文修正

讓他十分害怕  
只希望自己明年度別再這麼苦命了  
演藝娛樂產業加入積極轉型提升競爭ㄟ

# Introduction (cont'd)

- Imperfect acoustic models with phoneme loss.
- The finals of some characters are lost.



# Introduction (cont'd)

- Proposed methods:
  - Reconstruct the sentence by **language model**.

# Introduction (cont'd)

- For example, let  $Z = \text{演一 口樂 產一}$

$$W^* = \arg \max_W P(W | Z)$$

$$= \arg \max_W \frac{P(W)P(Z | W)}{P(Z)}$$

$P(Z)$  is independent of  $W$

$$= \arg \max_W P(W)P(Z | W)$$

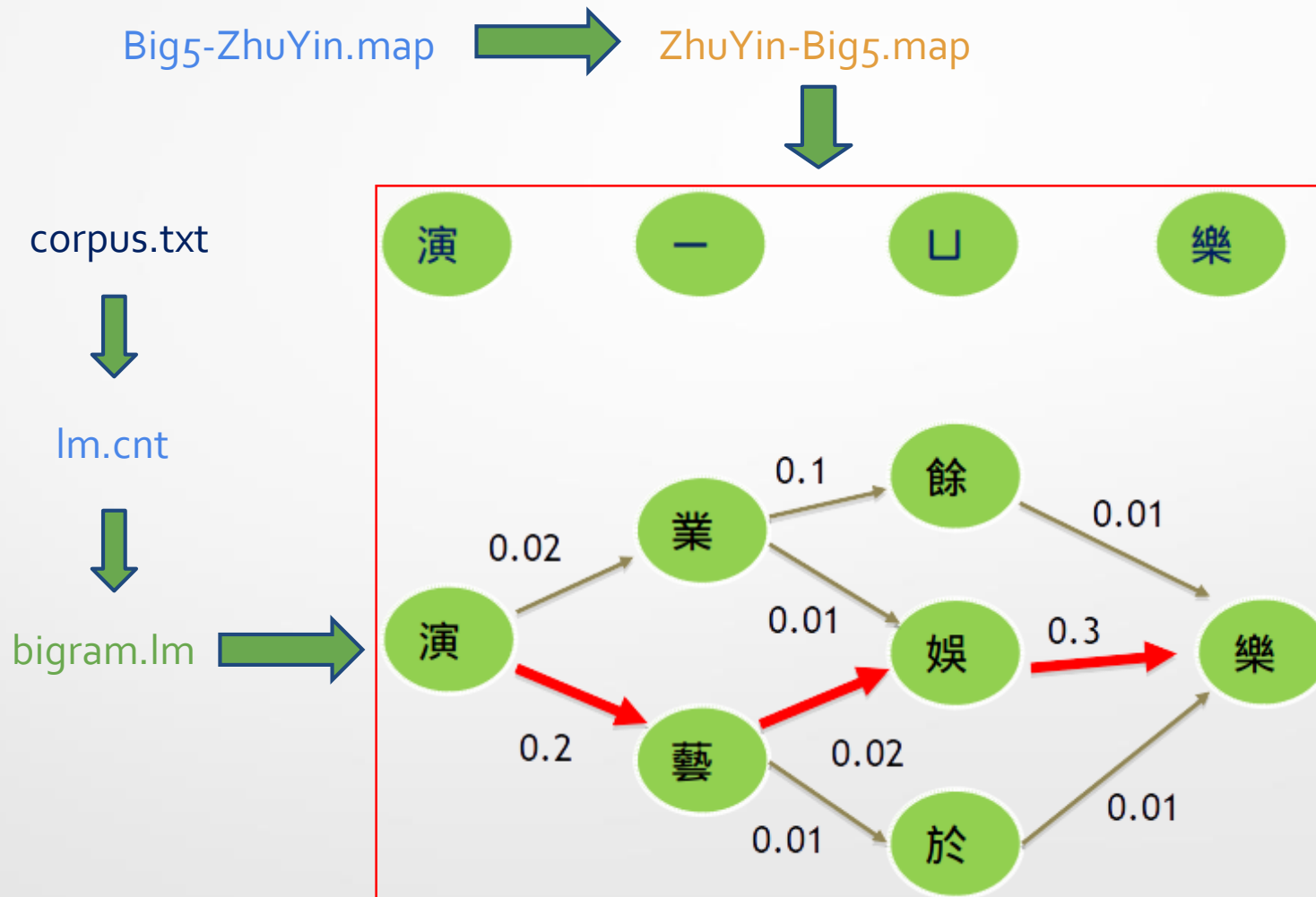
$W = w_1 w_2 w_3 w_4 \dots w_n, Z = z_1 z_2 z_3 z_4 \dots z_n$

$$= \arg \max_W \left[ P(w_1) \prod_{i=2}^n P(w_i | w_{i-1}) \right] \left[ \prod_{i=1}^n P(z_i | w_i) \right]$$

$$= \arg \max_{W, P(Z|W) \neq 0} \left[ P(w_1) \prod_{i=2}^n P(w_i | w_{i-1}) \right]$$

**Bigram language model**

# Flowchart





# SRILM

- SRI Language Model toolkit
  - <http://www.speech.sri.com/projects/srilm/>
- A toolkit for building and applying various statistical language models
- Useful C++ classes
- Using/Reproducing some of SRILM

# SRILM (cont'd)

- Build it from source code (Provided on course website)
  - Allows you to use SRILM library
- Or download the executable from the course website to finish the first part of HW<sub>3</sub>
  - Different platform:
    - i686 for 32-bit GNU/Linux
    - i686-m64 for 64-bit GNU/Linux (CSIE workstation)
    - Cygwin for 32-bit Windows with cygwin environment

# SRILM (cont'd)

- You are **strongly recommended** to read FAQ on the course website
- Possibly useful codes in SRILM
  - \$SRIPATH/misc/src/File.cc (.h)
  - \$SRIPATH/lm/src/Vocab.cc (.h)
  - \$SRIPATH/lm/src/ngram.cc (.h)
  - \$SRIPATH/lm/src/testError.cc (.h)

# SRILM (cont'd)

- Big5 Chinese Character separator written in perl:
  - perl separator\_big5.pl corpus.txt > corpus\_seg.txt
  - Why we need to separate it? (Use char or word?)

```
1 國民黨 立委 帶領 支持者 參加 升旗 心情 百感交集
2 多位 中國國民黨 籍 立法委員今天 一大早 帶領 支持者 到 總統府
3 在國民黨 失去 政權 後 第一次 參加 元旦 總統府 升旗典禮
4 有立委 感慨 國民黨不 團結 才會 失去 政權
5 有立委 則 猛 批 總統 陳水扁
6 人人 均 顯得 百感交集
7 國民黨籍立委 潘 維 剛 丁 守 中 蔡
8 到 總統府 前 參加 升旗典禮
9 潘 維 剛 表示
10 新世紀 的第一 天 參加 升旗典禮 讓
11 這一年來 政局 像 雲霄飛車 般 起伏
12 她 沒想到 政權 改變 影響 會 這麼
13 丁 守 中 表示
14 陳 總統 應該 立即 拿出 具體 政策
```



```
1 國 民 黨 立 委 帶 領 支 持 者 參 加 升 旗 心 情 百 感 交 集
2 多 位 中 國 國 民 黨 籍 立 法 委 員 今 天 一 大 早 帶 領 支 持 者 到 總 統 府
3 在 國 民 黨 失 去 政 權 後 第 一 次 參 加 元 旦 總 統 府 升 旗 典 禮
4 有 立 委 感 慨 國 民 黨 不 團 結 才 會 失 去 政 權
5 有 立 委 則 猛 批 總 統 陳 水 扁
6 人 人 均 顯 得 百 感 交 集
7 國 民 黨 籍 立 委 潘 維 剛 丁 守 中 蔡 家 福 關 沃 暖 洪 讀 李
8 到 總 統 府 前 參 加 升 旗 典 禮
9 潘 維 剛 表 示
10 新 世 紀 的 第 一 天 參 加 升 旗 典 禮 讓 她 百 感 交 集
11 這 一 年 來 政 局 像 雲 霄 飛 車 般 起 伏 不 知 何 時 能 落 地
12 她 沒 想 到 政 權 改 變 影 響 會 這 麼 大
13 丁 守 中 表 示
14 陳 總 統 應 該 立 即 拿 出 具 體 政 策 打 開 兩 岸 僵 局
```

# SRILM (cont'd)

- `./ngram-count -text corpus_seg.txt -write lm.cnt -order 2`
  - -text: input text filename
  - -write: output count filename
  - -order: order of ngram language model
- `./ngram-count -read lm.cnt -lm bigram.lm -unk -order 2`
  - -read: input count filename
  - -lm: output language model name
  - -unk: view OOV as <unk>. Without this, all the OOV will be removed

# Example

corpus\_seg.txt

在國民黨失去政權後第一次參加元旦總統府升旗典禮  
有立委感慨國民黨不團結才會失去政權  
有立委則猛批總統陳水扁  
人人均顯得百感交集

lm.cnt

夏 11210  
俸 267  
鵠 7  
祇 1  
微 11421  
檣 27  
.....

bigram.lm

\data\  
ngram 1=6868  
ngram 2=1696830

\1-grams:

-1.178429 </s>  
-99 <s> -2.738217  
-1.993207 一 -1.614897  
-4.651746 乙 -1.370091  
.....

(log probability)

(backoff weight)

## SRILM (cont'd)

- `./disambig -text testdata/xx.txt -map ZhuYin-Big5.map -lm bigram.lm -order 2 > $output`
  - `-text`: input filename, `xx = 1, 2, ..., 10`
  - `-lm`: input language model
  - `-map`: a mapping from (注音/國字) to (國字)
  - You should generate this mapping by yourself from the given Big5-ZhuYin.map.
  - Do not directly copy this command, please replace `xx.txt` with `1.txt~10.txt`.

# Generate Map

Big5-ZhuYin.map

一 一' / 一' / 一 \_  
乙 一 ^  
丁 ㄅ 一 ㄥ \_  
柒 ㄟ 一  
乃 ㄋ ㄣ ^  
玖 ㄐ 一 ㄣ ^  
...  
...  
長 ㄣ ㄣ ^ / ㄣ ㄣ ^  
行 ㄒ 一 ㄥ ^ / ㄣ ㄣ ^  
...



ZhuYin-Big5.map

ㄅ 八 匕 卜 不 卞 巴 比 丙 包 ...  
八 匕  
匕 匕  
卜 卜  
...  
...  
ㄣ 仆 匹 片 丕 叵 平 扒 扑 疋 ...  
仆 匹  
匹 匹  
...  
...

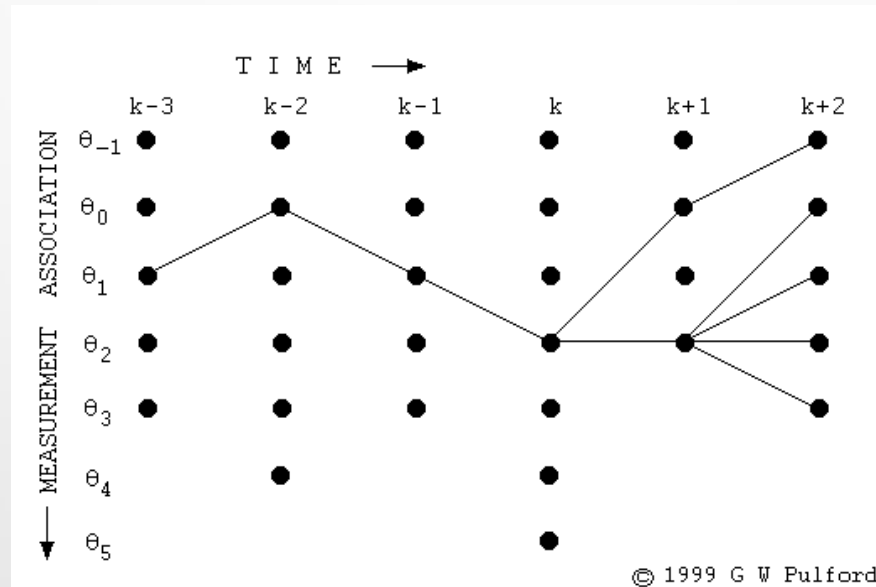


# Generate Map (cont'd)

- Be aware of polyphones(破音字)
- There could be arbitrary spaces between all characters.
- Key - value pairs
- Can be random permutation

# My Disambig

- Implement your version of disambig.
- Use dynamic programming (Viterbi).
- The vertical axes are candidate characters.



# Step by Step

- **Segment corpus and all test data into characters**
  - `./separator_big5.pl corpus.txt > corpus_seg.txt`
  - `./separator_big5.pl testdata/xx.txt > testdata/seg_xx.txt`
  - You should rename the segmented testdata as testdata/1.txt, testdata/2.txt... and use them in the following task.
- **Train character-based bigram LM**
  - Get counts:
  - `./ngram-count -text corpus_seg.txt -write lm.cnt -order 2`
  - Compute probability:
  - `./ngram-count -read lm.cnt -lm bigram.lm -unk -order 2`

## Step by Step (cont'd)

- **Generate ZhuYin-Big5.map from Big5-ZhuYin.map**
  - See FAQ 4
- **Using disambig to decode testdata/xx.txt**
  - `./disambig -text testdata/xx.txt -map ZhuYin-Big5.map -lm bigram.lm -order 2 > $output`
- **Using mydisambig to decode testdata/xx.txt**

# Tips

- **C++ is Required**
  - Speed
  - SRILM compatibility and utility
  - You must provide **Makefile** for execution (See. Grading Procedure for details)
- **Dual OS or VirtualBox with Ubuntu **strongly** recommended**
- **Your output format should be consistent with SRILM**
  - `<s>` 這是一個範例格式 `</s>`
  - There are an `<s>` at the beginning of a sentence, a `</s>` at the end, and whitespaces in between all characters.
  - Zero credit if your format is incorrect

# How to deal with Chinese char?

- Chinese character: You should use Big5 encoding
- All testing files are encoded in Big5
- A Chinese character in Big5 is always 2 bytes, namely, `char[2]` in C++

# Submission Example:

student ID: ro4922167

- ro4922167/ (see submit\_files\_template/ in dsp\_hw3.zip)
  - result1/1.txt~10.txt (generated from SRILM disambig with your LM by yourself)
  - your codes
  - Makefile
  - report.pdf
- Compress the folder to a zip file and upload it to Ceiba.
- 20% of the final score will be taken off for wrong format.

# Makefile

```
1 # The following two variable will be commandline determined by TA
2 # For testing, you could uncomment them.
3 SRIPATH ?= /data/DSP_HW3/103_2/srilm-1.5.10
4 MACHINE_TYPE ?= i686-m64
5 LM ?= bigram.lm
6
7 CXX = g++
8 CXXFLAGS = -O3 -I$(SRIPATH)/include -w --std=c++11
9 vpath lib%.a $(SRIPATH)/lib/$(MACHINE_TYPE)
10
11 TARGET = mydisambig
12 SRC = mydisambig.cpp
13 OBJ = $(SRC:.cpp=.o)
14 TO = ZhuYin-Big5.map
15 FROM = Big5-ZhuYin.map
16 .PHONY: all clean map run
17
18 all: $(TARGET)
19
20 $(TARGET): $(OBJ) -lloolm -ldstruct -lmisc
21     $(CXX) $(LDFLAGS) -o $@ $^
22
23 %.o: %.cpp
24     $(CXX) $(CXXFLAGS) -c $<
25
26 run:
27     @#TODO How to run your code toward different txt?
28     @for i in $(shell seq 1 10) ; do \
29         echo "Running $$i.txt"; \
30         ./mydisambig -text testdata/$$i.txt -map $(TO) -lm $(LM) -order 2 > result2/$$i.txt; \
31     done;
32
33 map:
34     @#TODO How to map?
35     @echo "Mapping!"
36     @./mapping $(FROM) $(TO)
37     @python mapping.py $(FROM) $(TO)
38     @sh mapping.sh $(FROM) $(TO)
39     @perl mapping.pl Big5-ZhuYin.map ZhuYin-Big5.map
40
41 clean:
42     $(RM) $(OBJ) $(TARGET)
```



# Report

- Your report should include:
  - Your environment (CSIE workstation, Cygwin, ...)
  - How to “compile” your program
  - How to “execute” your program
    - Not familiar with makefile is fine, tell me how to execute your program
    - However, you should also strictly follow the spec (regulations about filenames, input files and output files)
  - ex: `./program -a xxx -b yyy`
  - What you have done
  - NO more than two A4 pages.

# Grading Procedure

- There are some files provided by TA **but you shouldn't upload them**
  - Big5-ZhuYin.map, bigram.lm, testdata...
  - **Strictly follow regulations about format**
  - However, you can utilize the files in makefile
- **test\_env/** in dsp\_hw3.zip shows locations of files during evaluation
- In the following slides, this color specify makefile commands of evaluation scripts

# Grading Procedure (cont'd)

- (20%) You strictly follow format regulation
- Initialization
  - `make clean`
  - copy ta's bigram.lm, Big5-ZhuYin.map, testdata to your directory
- (10%) Your code can be successfully compiled
  - `make MACHINE_TYPE=i686-m64 SRIPATH=/home/ta/srilm-1.5.10 all`
  - i686-m64 is TA's platform
  - Your code should be machine-independent(system("pause") is invalid in my system) and the user can easily specify the platform and SRILM path

# Grading Procedure (cont'd)

- (10%) Correctly generate ZhuYin-Big5.map
  - `make map` (it should generate ro4922167/ZhuYin-Big5.map)
  - check if ro4922167/ZhuYin-Big5.map is correct
  - (You have to write your own makefile to achieve it. Generation must be based on ro4922167/Big5-ZhuYin.map)
  - (Your output in this step should be ro4922167/ZhuYin-Big5.map )
  - (python/perl/C/C++/bash/awk permitted)
- (20%) Correctly use SRILM disambig to decode ZhuYin-mixed sequence
  - Check if result1/1.txt~10.txt is the same as expected

# Grading Procedure (cont'd)

- (10%) mydisambig program can run with no errors and crashes
  - make MACHINE\_TYPE=i686-m64 SRIPATH=/home/ta/srilm-1.5.10 LM=bigram.lm run
  - (it should run based on bigram.lm and generate result2/1.txt~10.txt)
- (20%) Your results decoded by your own program are the same as expected
  - check result2/1.txt~10.txt
  - TA's testdata will be **segmented testdata**, not the given raw testdata. **DO NOT** use *"perl separator\_big5.pl"* in your makefile to separate testing data again.

# Grading Procedure (cont'd)

- (10%) Your report contains required information
- (10% bonus!) Your program can support trigram language models with speed pruning.
  - Write down how to execute your trigram code in your report.

# If there are runtime errors during TA's testing

- Like compilation error, crash...
  - TA will ask you to demo your program only with the files you uploaded.
  - If you can prove that you followed the rules correctly, you will get your credits.

# Late Penalty

- 10% each 24 hours, according to the **announced deadline** instead of the deadline on Ceiba
- 100 -> 90 -> 80, not 100 -> 90 -> 81



# Notes

- Follow the spec!!!!
- All of your program should finish the tasks assigned below 10 minutes
- Totally checking the correctness with good documents is YOUR JOB
- Only the latest files you uploaded to Ceiba will be evaluate (All of your previous uploaded version will be ignored)

# Reminders and Suggestions

- Read the spec carefully
- Finish the first part (SRILM disambig) as early as possible
  - If everything goes well, you should finish the first part in an hour
  - Fix the issue of dependencies early
  - Big5 encoding issue, **iconv** not recommended
- Be sure that you prepare the correct Makefile
  - Grading procedure is in part automatically done by scripts. You can see the details in the previous slides
- See the FAQ in the website

# Reminders and Suggestions

- Contact TA if needed
  - email : [ntudigitalspeechprocessingta@gmail.com](mailto:ntudigitalspeechprocessingta@gmail.com)  
title: [HW3] bxxxxxxx (your student number)
  - **Check** [email-FAQ!](#)
  - TA will not help you debug your program