

Digital Speech Processing Homework 3

2015/05/06

r03942039@ntu.edu.tw 呂相弘

Outline

- **Introduction**
- **SRILM**
- **Requirement**
- **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ㄟㄟ樂產ㄟㄟ入積ㄟㄟ型提ㄟ競爭ㄟ

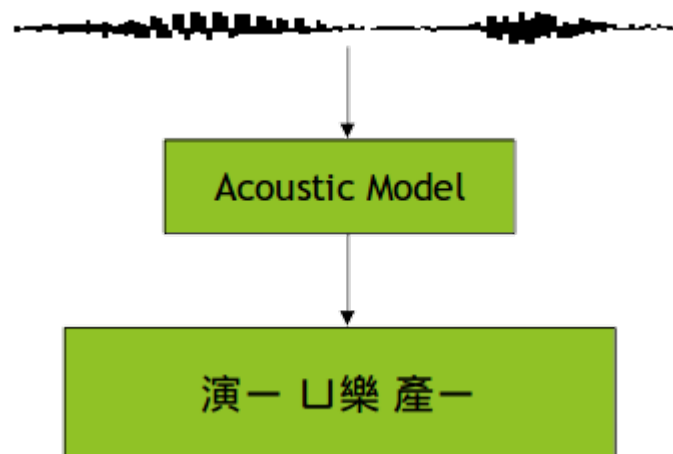
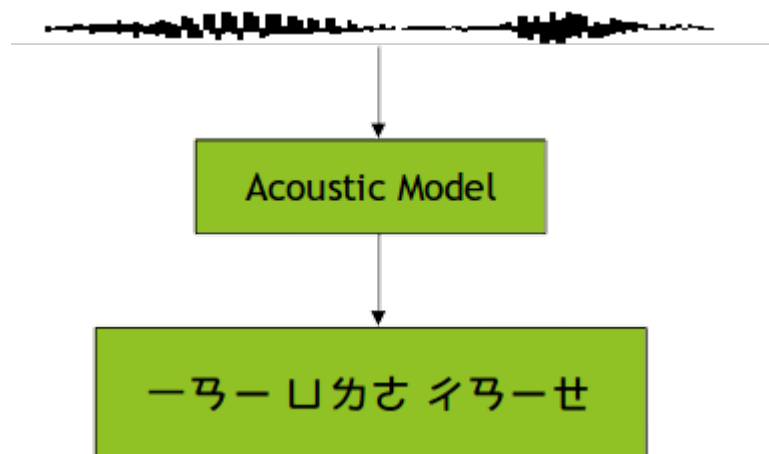


HW3: 注音文修正

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

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



Introduction

- **Proposed methods:**
 - Reconstruct the sentence by **language model**.
- **For example, let $Z = \text{演} \mid \text{山樂產} \mid$**

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

$$= \arg \max_W \frac{P(W)P(Z \mid W)}{P(Z)} \quad P(Z) \text{ is independent of } W$$

$$= \arg \max_W P(W)P(Z \mid W) \quad W=w_1w_2w_3w_4\dots w_n, Z=z_1z_2z_3z_4\dots z_n$$

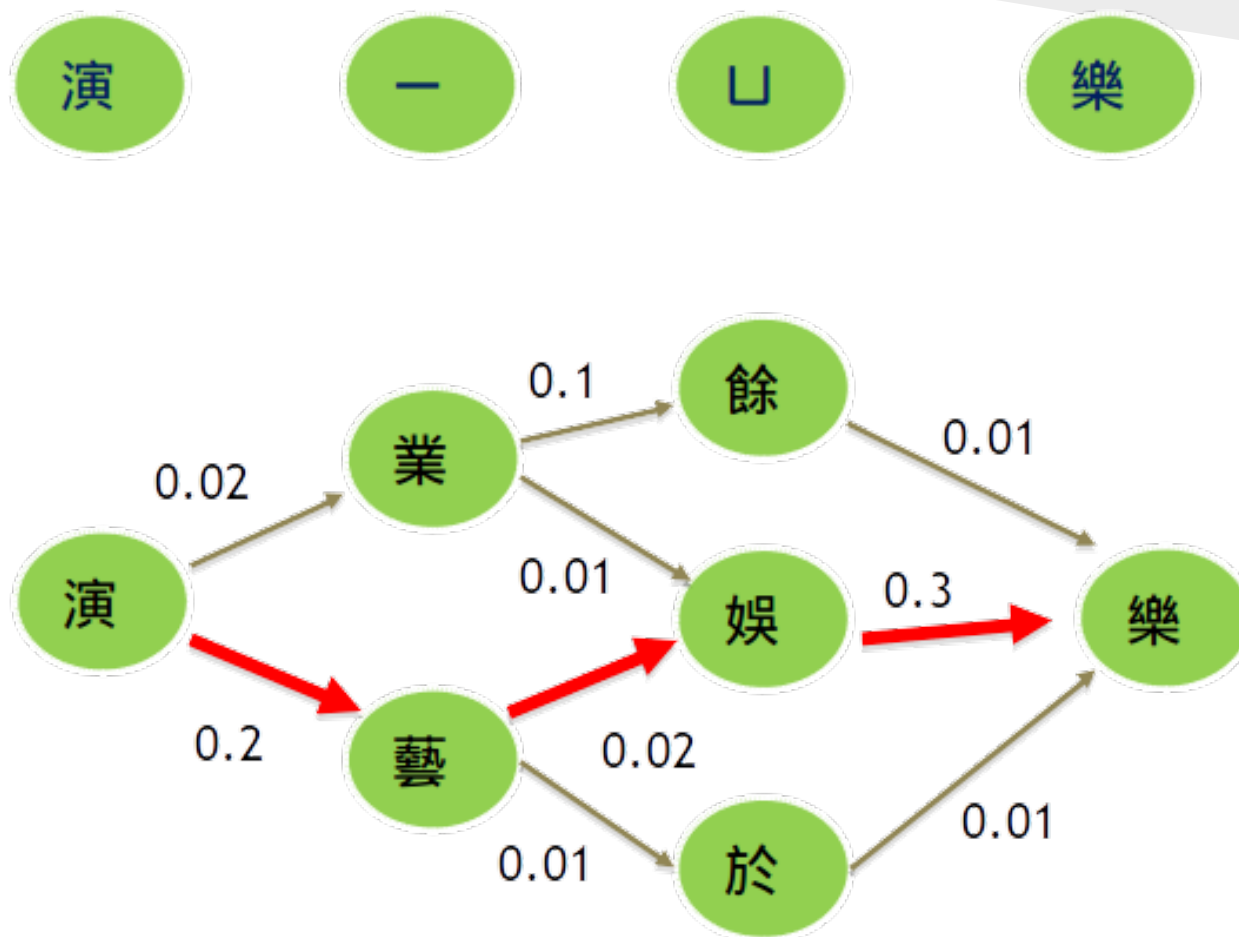
W 出現這個句子的機率 這個句子出現這個注音文的機率

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

不用看這個

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

Example



Goal

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

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

- **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


- **Download the executable from the course website**
 - **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**
- **Build it from source code with your own implementation.**

SRILM

- 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

- Big5 Chinese Character separator written in perl:
 - perl separator_big5.pl corpus.txt > **corpus_seg.txt**



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	1	國民黨	立委	帶領	支持者	參加	升旗	心情	百感交集			
8	到	總統府	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
9	潘維剛	3	在	國民黨	失去	政權	後	第一次	參加	元旦	總統府	升旗典禮	
10	新世紀的	4	有	立委	感慨	國民黨不	團結	才會	失去	政權			
11	這一年來	5	有	立委	則	猛	批	總統	陳水扁				
12	她沒想到	6	人人	均	顯得	百感交集							
13	丁守中	7	國民黨籍	立委	潘維剛	丁守中	蔡家福	關沃	暖洪	讀李			
14	陳總統	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

- `./ngram-count -text corpus_seg.txt -write lm.cnt -order 2`
給定corpus 算每個字出現的次數
 - -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`
用每個字出現的次數 算出language model
 - -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 **bigram.lm**

人人均顯得百感交集



lm.cnt

夏 11210

俸 267

鵠 7

祇 1

微 11421

檣 27

.....



(log probability)

\data

ngram 1=6868

ngram 2=1696830

\1-grams:

-1.178429 </s>

-99 <s> **-2.738217**

-1.993207 一 **-1.614897** (backoff weight)

-4.651746 乙 **-1.370091**

.....

SRILM

- `./disambig -text $file -map $map -lm $LM -order $order`
 - `-text`: input filename
 - `-map`: a mapping from (注音/國字) to (國字)
要解碼的字 注音文 一個注音會對應到哪些中文的map
 - `-lm`: input language model
language model
 - **DO NOT COPY-PASTE TO RUN THIS LINE**
 - **You should generate this mapping by yourself from the given Big5-ZhuYin.map.**
bigram的

SRILM

Big5-ZhuYin.map

一 一 ˊ / 一 ˋ / 一 _
乙 一 ˋ
丁 ㄉㄧㄣˋ _
柒 ㄘㄞˋ _
乃 ㄋㄞˇ ˋ
玖 ㄐㄞˋ 又 ˋ
...
...
長 ㄌㄞˊ ㄉㄞˊ ˊ / ㄘㄞˊ ˋ
行 ㄒㄞˊ ㄌㄞˊ ˊ / ㄆㄞˊ ˋ
...



ZhuYin-Big5.map

ㄅ ㄆ ㄇ ㄏ ㄉ ㄋ ㄅ ㄅ ㄅ ㄅ ...
ㄆ ㄆ
ㄇ ㄇ
ㄏ ㄏ
ㄉ ㄉ
...
...
ㄋ ㄅ ㄆ ㄇ ㄏ ㄉ ㄋ ㄅ ㄅ ㄅ ...
ㄅ ㄅ
ㄆ ㄆ
...
...

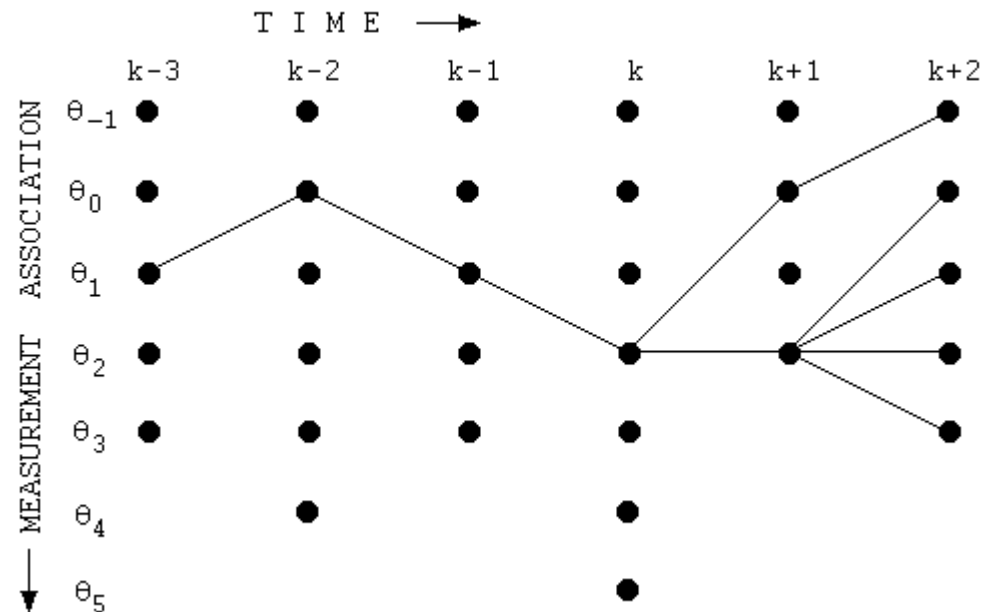
- Be aware of polyphones(破音字)
- There should be spaces between all characters.

Requirement I

- **Segment corpus and all test data into characters**
 - `./separator_big5.pl corpus.txt corpus_seg.txt`
 - `./separator_big5.pl <testdata/xx.txt> <testdata/xx.txt>`
- **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`
- **Generate the map from Big5-ZhuYin.map**
 - See FAQ 4
- **Using disambig to decode testdata/xx.txt**
 - `./disambig -text $file -map $map -lm $LM -order $order > $output`

Requirement II

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



Requirement II

- **You have to use C++**
 - Speed
 - SRILM compatibility and utility
 - you must provide Makefile.
- **Dual OS or VirtualBox with Ubuntu 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.

How to deal with Big5?

- All testing files are encoded in Big5
- A Chinese character in Big5 is always 2 bytes, namely, **char[2]** in C++

Submission

- When unzipped, your uploaded file should contain a directory as following:
 - hw3_[r03942039]/
 - ZhuYin-Big5.map (generated from provided Big5-ZhuYin.map by yourself)
 - result1/1.txt~10.txt (generated from SRILM disambig with your LM by yourself)
 - result2/1.txt~10.txt (generated from your disambig with your LM by yourself)
 - [your codes]
 - Makefile
 - Report[.pdf or .docx]

Submission

- The **report** should include:
 - Your environment (CSIE workstation, Cygwin, ...)
 - How to “compile” your program
 - How to “execute” your program (give me examples)
 - ex: `./program -a xxx -b yyy`
 - What you have done
 - **NO** more than two A4 pages.
 - **NO** “what you have learned”

Reminder

- Be sure that you prepare the correct Makefile
 - Grading procedure is in part automatically done by scripts. You can see the details in the following slides.
- See the **FAQ** in the website
- Contact TA if needed
 - r03942039@ntu.edu.tw 呂相弘

Grading

- (10%) Correctly generate ZhuYin-Big5.map
- (30%) Correctly use SRILM disambig to decode ZhuYin-mixed sequence.
- (10%) Your code can be successfully compiled.
- (10%) Your program can run with no errors and crashes.
- (20%) Your results decoded by your own program are the same as expected.
- (10%) Your report contains basic information.
- (10%) Your report is well-documented.
- **(10% bonus!)** Your program can support trigram language models with speed pruning.
- **(5% bonus!)** You implement other strategies trying to improve the results.

Grading Procedure

- When grading, TA will add additional data in **specific position**.
- hw3_[r03942039]/
 - ZhuYin-Big5.map
 - **Big5-ZhuYin.map** (provided but you shouldn't upload it)
 - **bigram.lm** (Don't upload your own language model)
 - **testdata/1.txt~10.txt** (segmented. This is provided but you shouldn't upload it)
 - result1/1.txt~10.txt
 - result2/1.txt~10.txt
 - [your codes]
 - Makefile
 - Report[.pdf or .docx]

Grading Procedure

- (10%) Correctly generate ZhuYin-Big5.map
 - check if **hw3_[r03942039]/ZhuYin-Big5.map** is correct
 - delete **hw3_[r03942039]/ZhuYin-Big5.map**
 - **make map** (it should generate hw3_[r03942039]/ZhuYin-Big5.map)
 - (You have to write your own makefile to achieve it. Generation must be based on **hw3_[r03942039]/Big5-ZhuYin.map**)
 - check if **hw3_[r03942039]/ZhuYin-Big5.map** is correct
 - python/perl/c++/c/matlab/bash/awk permitted
- (30%) Correctly use SRILM disambig to decode ZhuYin-mixed sequence.
 - check if result1/1.txt~10.txt is the same as expected.

Grading Procedure

- (10%) Your code can be successfully compiled.
 - **make MACHINE_TYPE=[TA's platform: i686-m64] SRIPATH=/home/r03942039/srilm-1.5.10 all**
 - Your code should be **machine-independent(system("pause") is invalid in my system.)** and the user can easily specify the platform and SRILM path.
- (10%) Your program can run with no errors and crashes.
- (20%) Your results decoded by your own program are the same as expected.
 - check result2/1.txt~10.txt
 - delete result2/1.txt~10.txt
 - **make LM=bigram.lm run** (it should run based on **bigram.lm** and generate result2/1.txt~10.txt)
 - check result2/1.txt~10.txt

Notes

- **Any incorrect format or naming error may lead to 0 credits.**
- **If the program cannot check your code with such error, any response is ignored.**
- **Totally checking the correctness with good documents is YOUR JOB.**

Makefile example

```
# The following two variable will be commandline determined by TA
# For testing, you could uncomment them.
SRIPATH ?= /data/DSP_HW3/103_2/srilm-1.5.10
MACHINE_TYPE ?= i686-m64
LM ?= bigram.lm

CXX = g++
CXXFLAGS = -O3 -I$(SRIPATH)/include -w
vpath lib%.a $(SRIPATH)/lib/$(MACHINE_TYPE)

TARGET = mydisambig
SRC = mydisambig.cpp
OBJ = $(SRC:.cpp=.o)
TO = ZhuYin-Big5.map
FROM = Big5-ZhuYin.map
.PHONY: all clean map run

all: $(TARGET)

$(TARGET): $(OBJ) -lloolm -ldstruct -lmisc
    $(CXX) $(LDFLAGS) -o $@ $^

%.o: %.cpp
    $(CXX) $(CXXFLAGS) -c $<

run:
    @#TODO How to run your code toward different txt?
    @for i in $(shell seq 1 10) ; do \
        echo "Running $$i.txt"; \
        ./mydisambig -text testdata/$$i.txt -map $(TO) -lm $(LM) -order 2 > result2/$$i.txt; \
    done;

map:
    @#TODO How to map?
    @echo "Mapping!"
    @#./mapping $(FROM) $(TO)
    @#matlab < mapping.m ;
    @#python mapping.py $(FROM) $(TO)
    @#sh mapping.sh $(FROM) $(TO)
    @#perl mapping.pl Big5-ZhuYin.map ZhuYin-Big5.map

clean:
    $(RM) $(OBJ) $(TARGET)
```