Digital Speech Processing
Homework 3

2015/05/06

r03942039@ntu.edu.tw 呂相弘

Outline

- **Introduction**
- **SRILM**
- **Requirement**
- **Submission and Grading**

Introduction

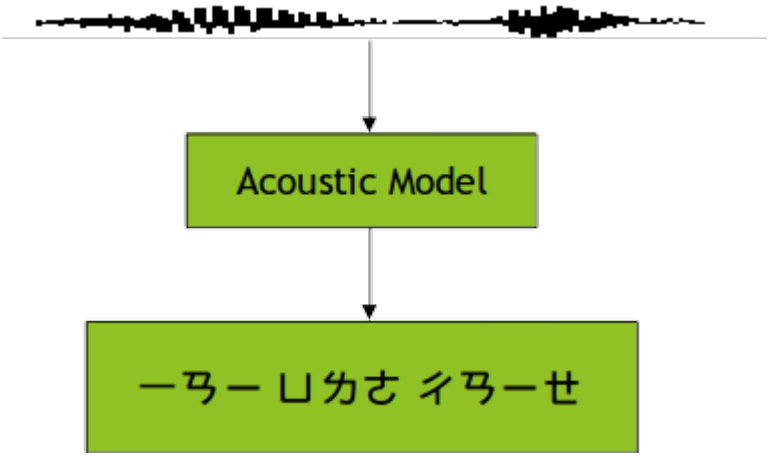
讓他十分ㄏ怕
只ㄒ望ㄗ己明ㄋ度別再這ㄇ ㄎ命了
演ㄧ ㄩ樂產ㄧ ㄐ入積ㄐ ㄓ型提ㄕ競爭ㄌ

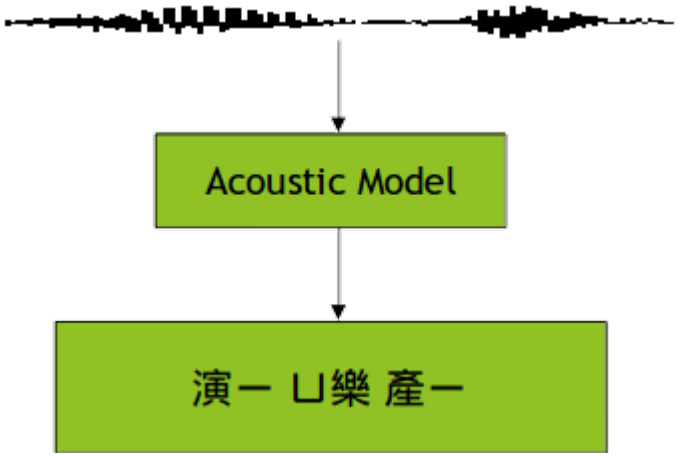
HW3:注音文修正

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力

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 = 演ㄧㄩ樂產ㄧ**

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

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

P(Z) is independent of W

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

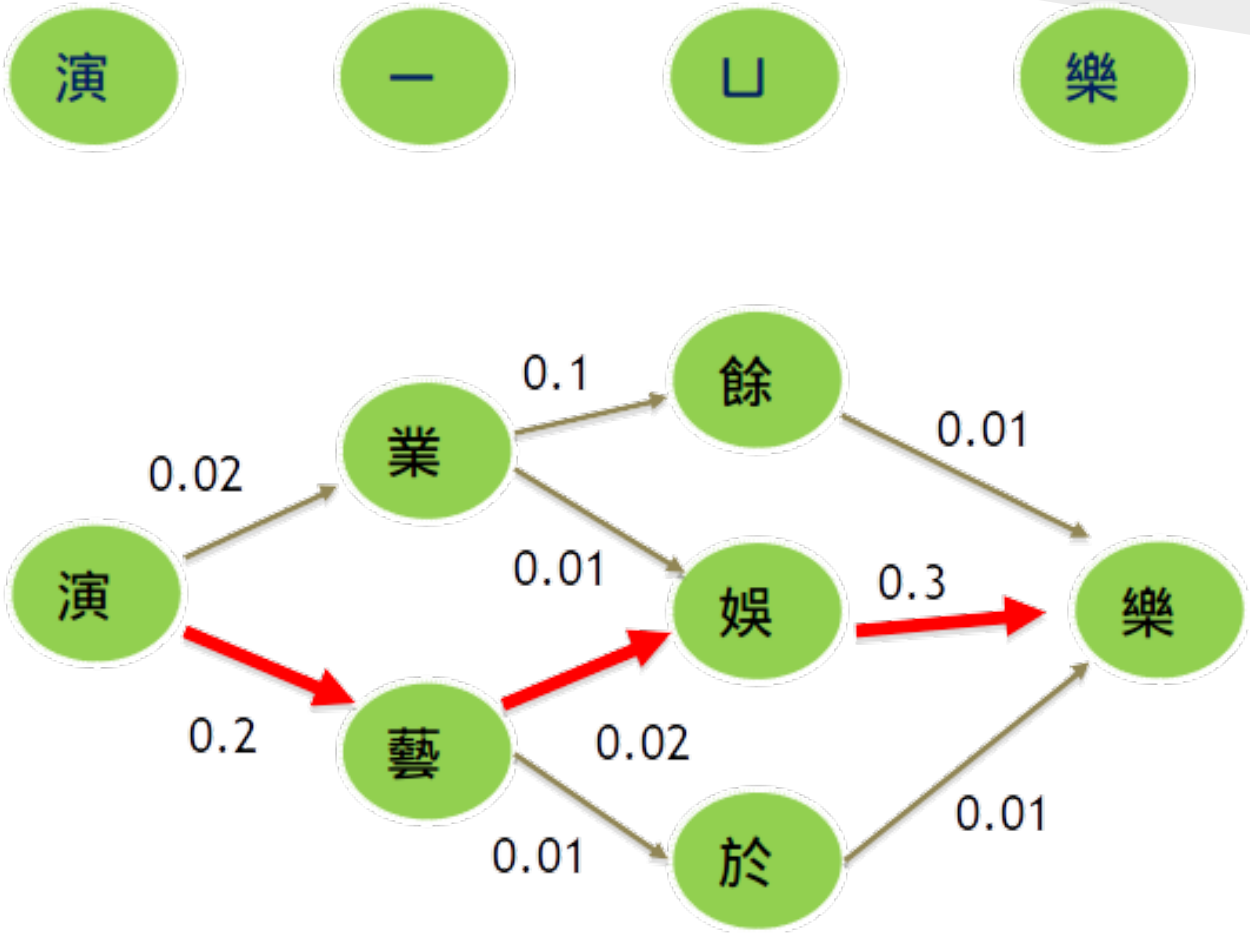
$W=w_1w_2w_3w_4...w_n$, $Z=z_1z_2z_3z_4...z_n$

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

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

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 黨籍立
8  到 總統府
9  潘 維 剛 表
10 新世紀 的第
11 這一 年來
12 她 沒想到
13 丁 守 中 表
14 陳 總統 應
```

```
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

- ./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
鴣	7
祇	1
微	11421
檎	27
……	

(log probability)

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
……
```

(backoff weight)

SRILM

- ./disambig –text $file –map $map –lm $LM –order $order
  - -text: input filename
  - -map: a mapping from (注音/國字) to (國字)
  - -lm: input language model
  - **DO NOT COPY-PASTE TO RUN THIS LINE**
  - **You should generate this mapping <u>by yourself</u> from the given Big5-ZhuYin.map.**

SRILM

Big5-ZhuYin.map

一 ㄧ ˊ/ ㄧ ˋ/ ㄧ _
乙 ㄧ ˇ
丁 ㄉㄧㄥ _
柒 ㄑㄧ _
乃 ㄋㄞ ˇ
玖 ㄐㄧㄡ ˇ
…
…
長 ㄔㄤ ˊ/ ㄓㄤ ˇ
行 ㄒㄧㄥ ˊ/ ㄏㄤ ˊ
…

ZhuYin-Big5.map

ㄅ 八 匕 卜 不 卞 巴 比 丙 包…
八 八
匕 匕
卜 卜
…
…
ㄆ 仆 匹 片 丕 回 平 扒 扑 疋…
仆 仆
匹 匹
…
…

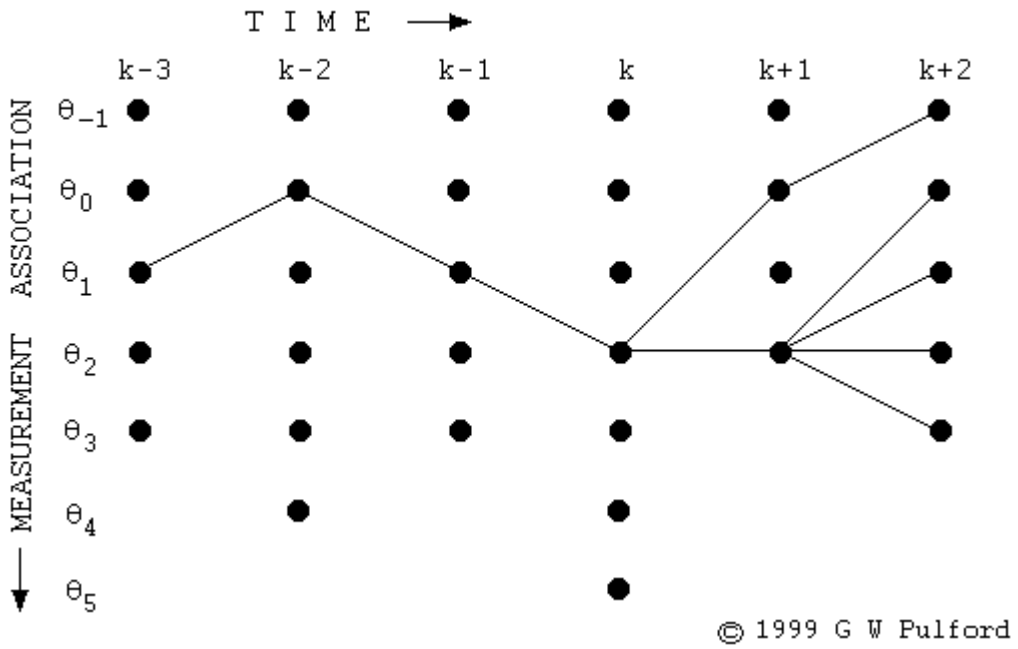
- **Be aware of polyphones(破音字)**
- **There should be spaces between all characters.**

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 <u>a directory as following:</u>
  - 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

```makefile
# 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) -loolm -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)
```