

Digital Speech Processing

Homework #3

ZhuYin Decoding

曾元 (Modified from 陳建成 DSP2020 ver.)

November 9, 2022

Due on 23:59, December 2, 2022

Outline

1. Introduction
2. SRILM Toolkit
3. Homework
4. Requirements
5. Grading
6. Contact TAs

Introduction

ZhuYin is used widely in our daily life.

- 勺完飯要厂珍奶買乚排口？ → 吃完飯要喝珍奶買雞排嗎？
- 我覺勿助教真勿很嚴格 → 我覺得助教真的很嚴格
- 老尸我先走勿，勺勺 → 老師我先走了，掰掰
- 匕虫是勺以都勺喝水勿 → 肥宅是可以都不喝水的

(Some sentences are combination of real examples.)

In speech recognition, imperfect acoustic models with phoneme loss yield similar result, which we have to correct by post-processing.



→ 「演藝娛樂產業」

Sentences can be reconstructed with the help of **language model**.

$$W^* = \operatorname{argmax}_W P(W | Z) \quad (1)$$

$$= \operatorname{argmax}_W \frac{P(W) P(Z | W)}{P(Z)} \quad (2)$$

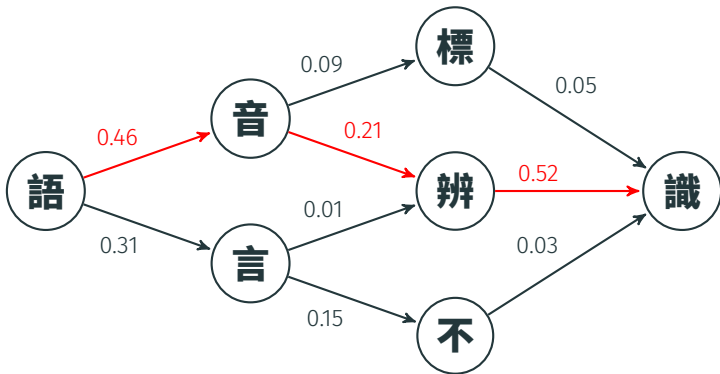
$$= \operatorname{argmax}_W P(W) P(Z | W) \quad (3)$$

$$= \operatorname{argmax}_W [P(w_1) \prod_{i=2}^n P(w_i | w_{i-1})] [\prod_{i=1}^n P(z_i | w_i)] \quad (4)$$

$$= \operatorname{argmax}_{W, P(Z|W) \neq 0} [P(w_1) \prod_{i=2}^n P(w_i | w_{i-1})] \quad (5)$$

Viterbi Algorithm

Viterbi algorithm gives the path with the greatest probability.
For example, 語一_レ識



SRILM Toolkit

- SRILM Toolkit stands for SRI Language Modeling Toolkit.
- It is a toolkit for building and applying various statistical language models.
- It provides useful C++ libraries, which we are going to exploit in this homework.
- You can either compile from source code on your own or download our pre-built docker image (**recommended!**).

We provide a pre-built docker image, and **strongly recommend** you complete hw3 in this environment, in order to allow us to automatically execute and grade your program. If you still choose to compile from source code, please confirm you can run your code on our environment.

(Ubuntu 18.04, GCC version 7.4.0)

For pre-built docker image, simply use the following command¹:

```
docker run -it ntudsp2020autumn/srilm
```

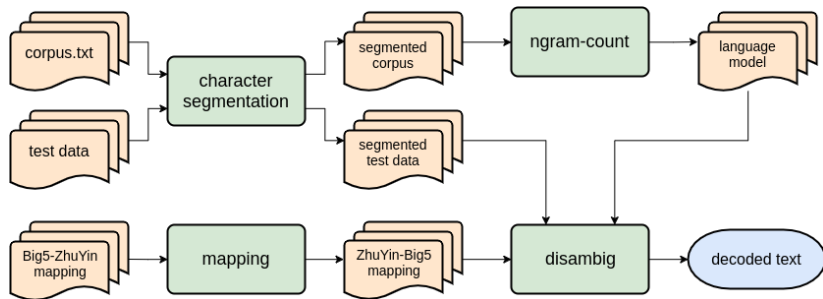
As for compiling on your own, please refer to FAQ².

¹You can use the `-v` option to mount folders, and `--name` to name your container

²<http://speech.ee.ntu.edu.tw/DSP2022Autumn/hw3/FAQ.html>

Homework

Workflow



Provided Files

Big5-ZhuYin.map

- a mapping from Chinese character to ZhuYin

corpus.txt

- corpus data

Makefile

- Makefile template

separator_big5.pl

- script used to separate Chinese words

setup.sh

- script used to activate SRILM environment

test_data

- directory for test data

dsp_hw3

```
|— Big5-ZhuYin.map
|— corpus.txt
|— Makefile
|— separator_big5.pl
|— setup.sh
└— test_data
   └— ...
```

To activate SRILM environment, simply use the provided script:

```
source setup.sh
```

You have to modify *SRILM_REP_PATH* and *MACHINE_TYPE* on your own if not using provided docker image.

Character Segmentation

Separate Chinese words into characters:

```
perl separator_big5.pl $1 > $2
```

\$1 input text file

\$2 output segmented file

- 魔法科高中的劣等生 → 魔法 科 高中 的 劣 等 生
- 大家都很有進展 → 大 家 都 很 有 進 展
- 數位語音處理概論 → 數 位 語 音 處 理 概 論

Language Model

Build the language model:

```
ngram-count -text \$1 -write \$2 -order \$3
```

\$1 input text file

\$2 output count file

\$3 order of n-gram

```
ngram-count -read \$1 -lm \$2 -order \$3 -unk
```

\$1 input count file

\$2 output language model file

\$3 order of n-gram

unk view OOVs as <unk> instead of removing them

- count file

```
夏 11210
俸 267
鳩 7
祇 1
微 11421
```

- language model file

```
\data\  
ngram 1 = 6868  
ngram 2 = 1696830  
\1-grams:  
-1.178429 </s>  
-99 <s> -2.738217  
-1.993207 — -1.614897
```

ZhuYin-Big5 Map

Create a ZhuYin-Big5 mapping from Big5-ZhuYin mapping:

Big5-ZhuYin	ZhuYin-Big5
一 一` / 一` / 一_	ㄅ 八 匕 卜 ...
乙 一˘	八 八
丁 ㄉ一ㄥ_	匕 匕
...	...
長 ㄉㄨㄤˊ / ㄉㄨㄤ˘	ㄉㄨ ㄅ 匹 片 丕 ...
行 ㄒ一ㄥˊ / ㄒㄨㄥˊ	ㄅ ㄅ
...	...

- Be aware of **polyphones** (破音字).
- Do not forget Big5-Big5 mapping, e.g. 八八, 匕匕
- The first and second columns should be separated with **one tab**.
- Each observation should be separated with **one space**.
- You should allow arbitrary order of entries. The order of entries **SHOULD NOT** affect the output of your *disambig* program

Decode the data by:

1. disambig provided by SRILM

```
disambig -text \$1 -map \$2 -lm \$3 -order \$4 > \$5
```

\$1 segmented file to be decoded

\$2 ZhuYin - Big5 mapping

\$3 language model

\$4 order of language model

\$5 output file

2. your own disambig (MyDisambig)

- Please implement it with the Viterbi algorithm.
- Bigram decoding is required while implementing trigram decoding will score bonus points.
- Details will be described in *Requirements*.

Requirements

Your programs should be compiled by Makefile with:

```
make SRIPATH=$1 MACHINE_TYPE=$2
```

\$1 SRILM path

\$2 machine type

In provided docker image, *SRIPATH* is */root/srilm-1.5.10* and *MACHINE_TYPE* is *i686-m64*.

At least one executable named *mydisambig* should be compiled.

Mapping Program Format

You are allowed to use **C**, **C++**, or **Python** in this part. TA will run your program by the following command:

```
make map FROM=$1 TO=$2
```

\$1 Big5-ZhuYin mapping file (input)

\$2 ZhuYin-Big5 mapping file (output)

If you are using **C** or **C++**, your mapping program should be compiled in this or previous step.

MyDisambig Format

You are required to implement your own disambig using C++. It will be executed by the following command:

```
./mydisambig $1 $2 $3 $4
```

\$1 segmented file to be decoded

\$2 ZhuYin-Big5 mapping

\$3 language model

\$4 output file

The output format of your program should be **exactly the same** as that of SRILM disambig.

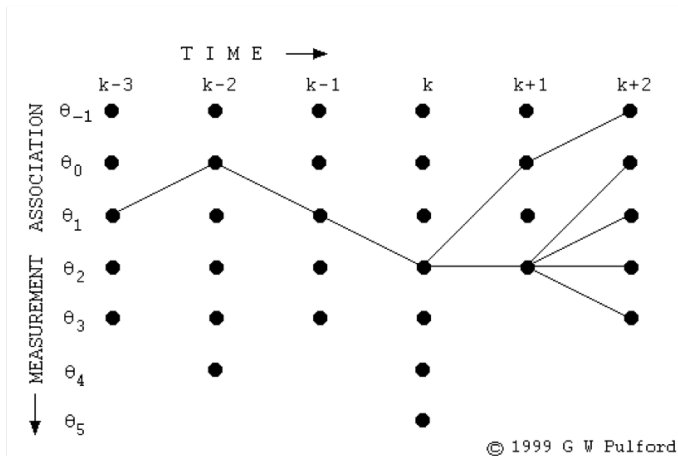
Do not hard-code paths in your source code.

Do not change the number or the order of the command-line arguments.

Hint for MyDisambig

Use dynamic programming (Viterbi).

The vertical axes are candidate characters.



Here is an example of SRILM disambig output:

```
<s> 保 持 社 交 距 離 </s>
```

- There is always a `<s>` at the beginning and a `</s>` at the end.
- There is exactly one space between each token.

- Using source code from the following libraries is **strongly forbidden**:
 - `$SRIPATH/include/VocabMap.h`
 - Source codes of *disambig* and Viterbi algorithm.
- But these SRILM libraries provide useful utilities that you may use:
 - `$SRIPATH/include/File.h`
 - `$SRIPATH/include/Ngram.h`
 - `$SRIPATH/include/Vocab.h`
- If you need other libraries, please contact us to ask for permission.

- A Chinese character in Big5 is always encoded with 2 bytes, namely, *char[2]* in *C++*.
- Be careful not to change the encoding of data, otherwise your program may fail to parse Big5 encoded files when grading.

Write a report (at most **two pages**) in **PDF** format, name it *report.pdf* and submit with your source code.

Your report should at least include the following contents:

- your name and student ID
- what you observed (e.g., disambig vs. MyDisambig)
- what you have done (e.g., trigram decoding)

If you implement trigram decoding, you should describe it in a detailed manner. The bonus points will be granted according to both program performance and your description.

Submission Format

Your submission should be a **ZIP** file and be uploaded to NTU Cool. Its file structure should be the same as the following:

```
<zip_file>
└─ hw3_<student_id>
    └─ Makefile
    └─ report.pdf
    └─ inc
        └─ [*h, *.hpp]
    └─ src
        └─ [*c, *.cc, *.cpp, *.py]
```

- *Makefile*, *report.pdf*, *inc/*, and *src/* are **MUST HAVE**, even if you don't have any header files.
- All of your source code files must be placed under *inc* and *src*.
- **DO NOT** create any subdirectory in *inc* and *src*.
- *<student_id>* should be in lowercase, e.g. *r01234567*.

Grading

Grading Method & Time Limits & Environment

Your mapping and disambig program will be tested respectively.

Mapping is allowed to run for **30 seconds** while MyDisambig is required to finish in **1 minute** for each input data. Programs will be interrupted when the time limit is exceeded.

For those who want to get bonus points, your trigram Mydisambig must also finish within **1 minute**.

The environment TA will use is the same as the provided docker image except for **different** SRILM path.

Grading Policy

$$\text{SCORE} = (\text{F} + \text{M} + \text{Mapping} + \text{MyDisambig} + \text{Report} + \text{Trigram})$$

Item	Score	Description
F (File Format)	0%	10 points will be deducted for incorrect file format.
M (Make)	0%	10 points will be deducted if we cannot execute your Makefile successfully.
Mapping	15%	Correctly generating mapping in time will receive full credits, otherwise you will get 0.
MyDisambig	70%	35% for successful execution in time, and 35% for accuracy.
Report	15%	You will get 0 if it is more than 2 pages.
Trigram	10%	Bonus for trigram MyDisambig.

Late Submission

Due on 23:59, December 2, 2022

You are still allowed to submit after the due date. The penalty for late submission is an exponential decay with decay rate 1.5%³ of the maximum grade applicable for the assignment, for each hour that the assignment is late.

An assignment submitted more than 3 days after the deadline will have a grade of zero recorded for that assignment.

$$\text{SCORE}_{\text{final}}(\text{hr}) = \begin{cases} \text{SCORE}_{\text{original}} \times 0.985^{\text{hr}} & , \text{hr} \leq 72 \\ 0 & , \text{hr} > 72 \end{cases}$$

³less than 70% after 24 hrs, 48% for 48 hrs and 33% for 72 hrs

Any form of cheating, lying, or plagiarism will not be tolerated.

Should you have any question or need help,

- please read the FAQ⁴ first.
- send email to *ntudsp2022ta@gmail.com* with “[HW3]” as the prefix of the email subject line.

曾元	Tue.	14:30 - 16:30
林棋竣	Tue.	9:00 - 11:00

Office hours

TA 林棋竣 will answer environment-related questions.

⁴<http://speech.ee.ntu.edu.tw/DSP2022Autumn/hw3/FAQ.html>