Digital Speech Processing
Homework 3

Nov 21 2018

許博竣

To complete the homework, you need to…

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

Outline

- Introduction
- SRILM
- Step by Step
- 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ㄌ

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 = 演一 ㄩ樂 產一

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

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

P(Z) is independent of W

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

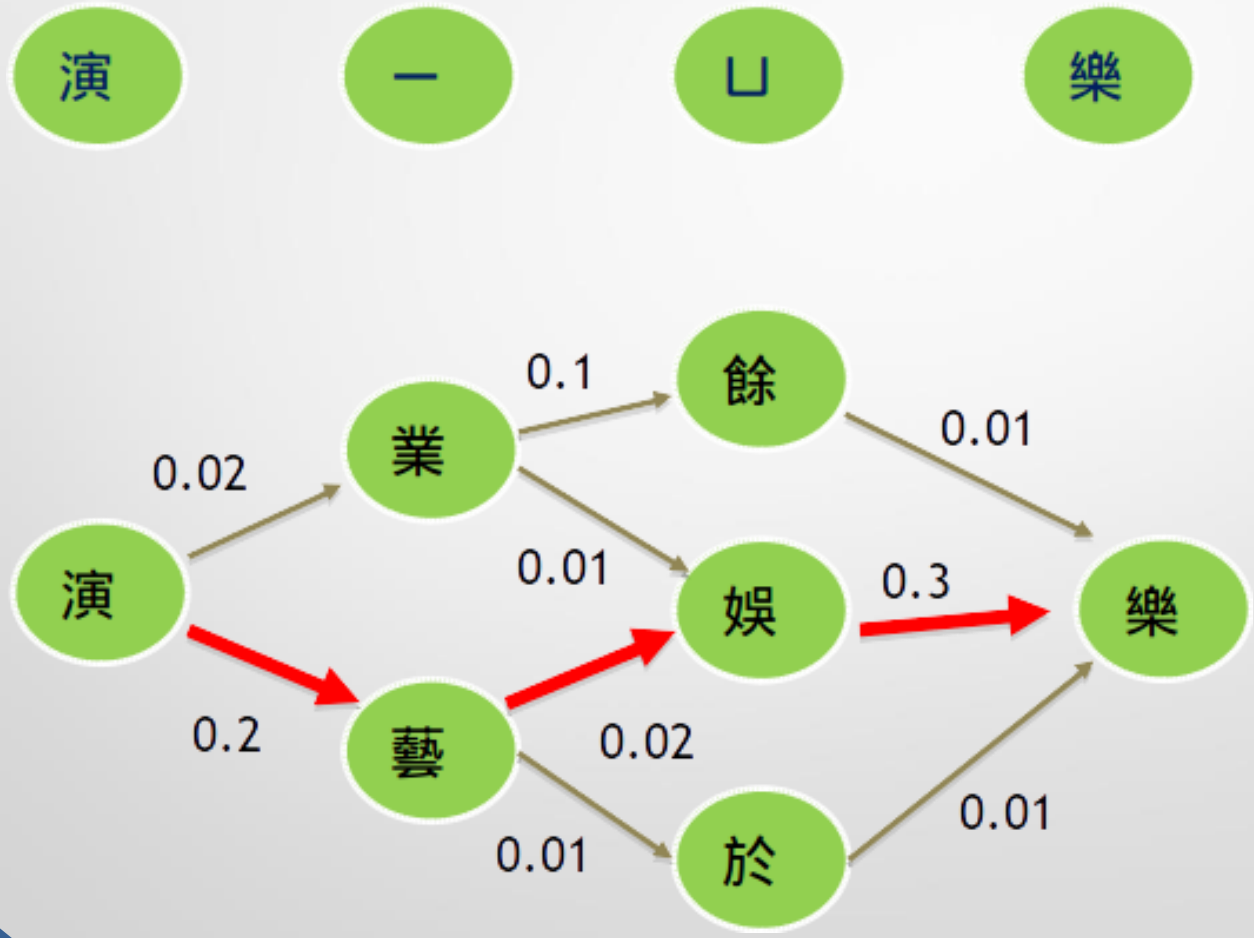
$W=w_1w_2w_3w_4...w_n$, $Z=z_1z_2z_3z_4...z_n$

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

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

Bigram language model

Example



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

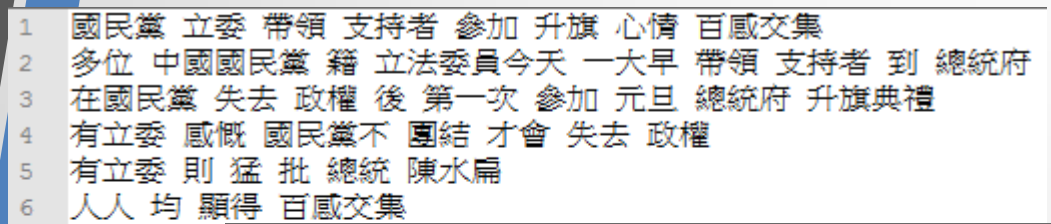
- Build it from source code (Provided on course website)
  - Allows you to use SRILM library
- Or download the executable from the course website to finish the first part of HW3
  - Different platform:
    - i686 for 32-bit GNU/Linux
    - i686-m64 for 64-bit GNU/Linux (CSIE workstation)
    - Cygwin for 32-bit Windows with cygwin environment

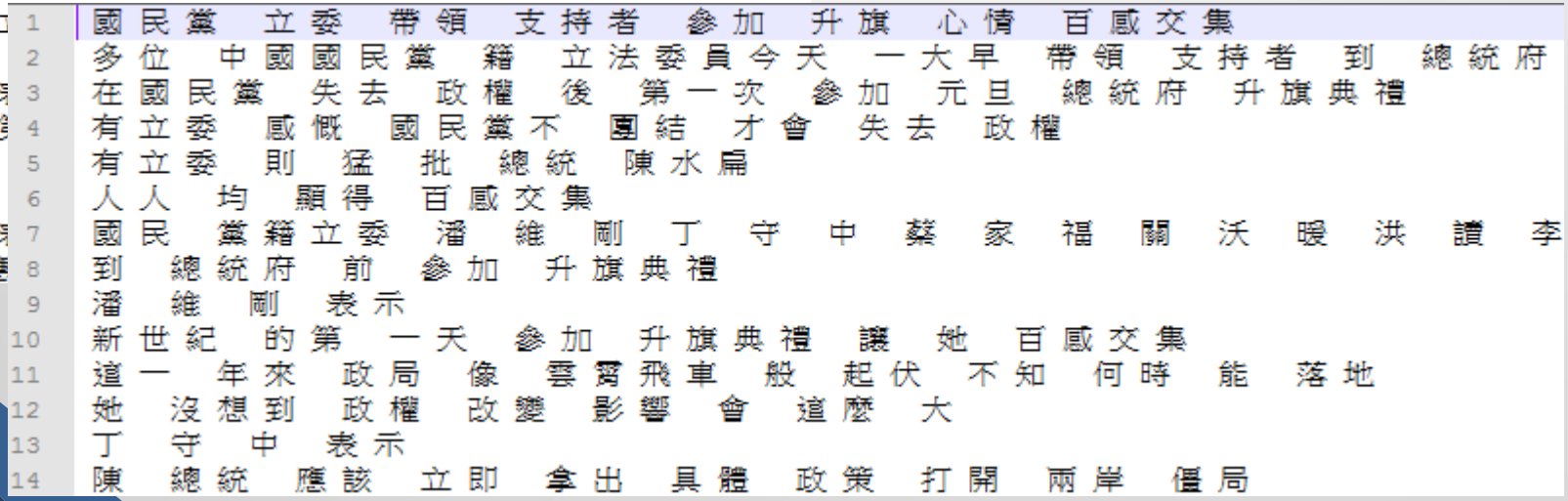
SRILM

- 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
  - Why we need to separate it? (Use char or word?)





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
鴿	7
祇	1
微	11421
檥	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)

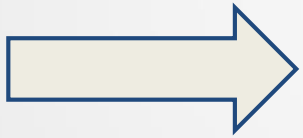
SRILM

- ./disambig –text $file –map $map –lm $LM –order $order
  - -text: input filename
  - -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 COPY-PASTE TO RUN THIS LINE!

Big5-ZhuYin -> ZhuYin-Big5

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 叵 平 扒 扑 疋…
仆 仆
匹 匹
…
…

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

Step by Step

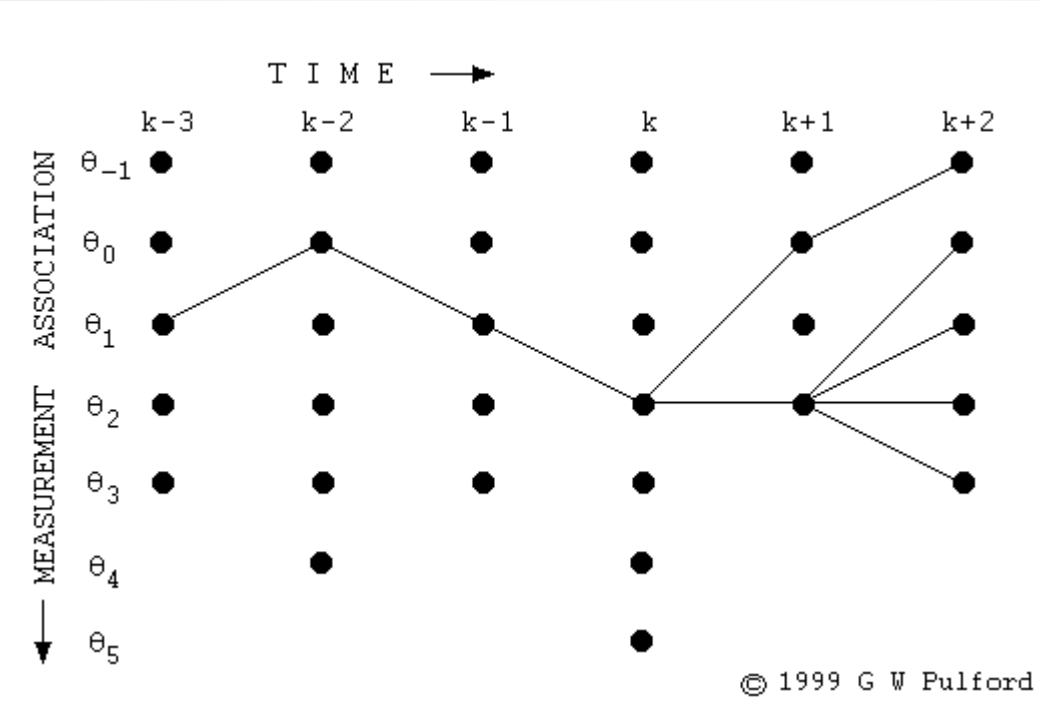
- Segment corpus and all test data into characters
  - ./separator_big5.pl corpus.txt >corpus_seg.txt
  - ./separator_big5.pl testdata/xx.txt >testdata/seg_xx.txt
  - You should rename the segmented testdata as testdata/1.txt, testdata/2.txt… and use them in the following task
- Train character-based 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
- Generate ZhuYin-Big5.map from Big5-ZhuYin.map
  - See FAQ 4

SRILM disambig

- Using disambig to decode testdata/xx.txt
  - ./disambig -text $file -map $map -lm $LM -order $order > $output

My Disambig

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



Tips

- **C++ is Required**
  - Speed
  - SRILM compatibility and utility
  - You must provide **Makefile** for execution

(See. Evaluation 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: r04922167

- dsp_hw3_r04922167.zip
- When unzipped, your uploaded file should contain a directory as following:
  - dsp_hw3_r04922167/
    - result1/1.txt~10.txt (generated from SRILM disambig with your LM by yourself)
    - your codes
    - Makefile
    - report.pdf
- Don't hw3_R04922167, HW3_r04922167, hw3_r04922167/Result1, hw3_r04922167/best_result1, hw3_r04922167/result1/segmented_1.txt…

Submission

- Your report should include:
  - Your environment (CSIE workstation, Cygwin, …)
  - How to "compile" your program
  - How to "execute" your program
    - You should 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.

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.

Grading

- (10%) Your code can be successfully compiled
- (10%) Correctly generate ZhuYin-Big5.map
- (30%) Correctly use SRILM disambig to decode ZhuYin-mixed sequence
- (10%) mydisambig program can run with no errors and crashes
- (25%) Your results decoded by your own program are the same as expected
- (10%) Your report contains required information
- (5%) You strictly follow format regulation
- (10% bonus!) Your program can support trigram language models with speed pruning.

Evaluation Procedure

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

Evaluation Procedure

- 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
- (10%) Correctly generate ZhuYin-Big5.map
  - make map (it should generate hw3_r04922167/ZhuYin-Big5.map)
  - check if hw3_r04922167/ZhuYin-Big5.map is correct
  - (You have to write your own makefile to achieve it. Generation must be based on hw3_r04922167/Big5-ZhuYin.map)
  - (Your output in this step should be hw3_r04922167/ZhuYin-Big5.map)
  - (python/perl/C/C++/bash/awk permitted)

Evaluation Procedure

- (30%) Correctly use SRILM disambig to decode ZhuYin-mixed sequence
  - Check if result1/1.txt~10.txt is the same as expected
- (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)
- (25%) 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

Late Penalty

- Deadline: 2018/12/14 (Fri.) 23:59:59
- 10% each 24 hours, according to the announced deadline instead of the deadline on Ceiba
- 100 -> 90 -> 80, not 100 -> 90 -> 81
- Submission after 12/16 (Sun.) 23:59:59 will get zero point

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
- Be sure that you prepare the correct Makefile
  - Evaluation procedure is in part automatically done by scripts. You can see the details in the previous slides
- See the FAQ in the website
- Contact TA if needed
  - Check email-FAQ!
  - TA will not help you debug your program