學號：R07942089

姓名：劉廷緯

系級：電信一

# DSP

# Homework 2 – Character-Based Language Model

## Environment

- < Ubuntu 6.4.0-17 >
  - Computer Architecture: x86_64
  - CPU op-mode(s): 32-bit, 64-bit
- < SRILM 1.5.10 >
- < g++ [gcc version 8.2.0 (GCC)] > (Tested)
- < g++ [gcc version 6.4.0 (GCC)] > (Tested)
- < g++ [gcc version 4.2.1 (GCC)] > (Tested)

## Usage

- Compile code:
  `$ make all`

- Separate training and testing data into separate characters:
  `$ make separate`

- Build Zhu-Yin to char mapping:
  `$ make map`
  This generates 2 files: I) ZhuYin-Big5.map, and II) ZhuYin-Utf8.map where:
  - I) ZhuYin-Big5.map: the Zhu-Yin to Chinease character mapping in big5 encoding
  - II) ZhuYin-Utf8.map: the Zhu-Yin to Chinease character mapping in big5 encoding for user verification in ordinary linux environment

- Build language model:
  `$ make build_lm`

- Decode with SRILM disambig:
  `$ make run_disambig`

- Decode with MY disambig:
  `$ make run`

- Decode with MY disambig but show output on screen instead of write to file:
  `$ make run_cout`

- Clean executables:
  `$ make clean`

- Clean everything generated in the above steps:
  `$ make cleanest`

- The variables `SRIPATH` and `MACHINE_TYPE` can be specified by the user at run time through the make command.

## What have I done

- First I've read the useful codes in SRILM, including:
  $SRIPATH/lm/src/LM.h
  $SRIPATH/misc/src/File.h
  $SRIPATH/lm/src/Prob.h
  $SRIPATH/lm/src/Ngram.h
  $SRIPATH/lm/src/Vocab.h
  $SRIPATH/lm/src/VocabMap.h
  $SRIPATH/lm/src/VocabMultiMap.h

- With the useful classes and functions of the SRILM, I implement my own disambig, a viterbi-based decoding process of the language model. Given a ZhuYin-mixed sequences obtained from an imperfect acoustic models with phoneme loss, reconstruct and decode the correct sentence using a character-based language model, in which the implemented mydisambig.cpp handles the decoding and reconstruction process.