DSP HW3

b06902048 資工二 李峻宇

Environment :

CSIE Workstation

Language :

Mapping : python3

mydisambig : C++

How to compile and execute :

compile :

```
make clean  //remove some file
make MACHINE_TYPE=i686-m64 SRIPATH=$srilm_path all  //compile mydisambig.cpp
```

execute :

```
make map  // execute mapping.py and generate ZhuYin-Big5.map
make MACHINE_TYPE=i686-m64 SRIPATH=$srilm_path LM=$bigram_path run //run
mydisambig
```

And the answer will be in `result2/`

What Have I Done :

mapping.py :

Write a python3 code to open `Big5-ZhuYin.map` with `encoding=big5hkscs`, and read it one line by one line. Use powerful python dictionary, as ZhuYin being key and Big5 being value. Finally we output it to another file `ZhuYin-Big5.map` .

mydisambig.cpp :

Firstly, make sure that the number of parameter and each parameter are correct, and I assume that each parameter will not be longer than 90.

Secondly, we read `ZhuYin-Big5.map` and `bigram.lm` respectively, then we use c++ map class to deal with `ZhuYin-Big5.map` and function in `Ngram.h` and `Vocab.h` to. deal with `bigram.lm` .

Finally, we read `/testdata/1~10.txt` one line by one line ane run a Viterbi algorithm to find the best path to fit, including counting probabilities between near two word and backtracking.