DSP HW3 Report

b08902045 資工四 袁紹奇

Environment

- CSIE workstation
- Use the docker image TA provided
- Mapping: python3
- mydisambig: C++17

How to run the code

Use the instruction TA provided to activate the docker environment. Run the following code to generate the segemented corpus file, and the language model. Then generate the mapping file from ZhuYin to Big5.

```
$ perl separator_big5.pl corpus.txt > corpus_seg.txt
$ ngram-count -text corpus_seg.txt -write lm.cnt -order 2
$ ngram-count -read lm.cnt -lm bigram.lm -order 2 -unk
$ make map
$ make all
```

Generate the segmented test data, and run `mydisambig` to generate the prediction of each test_data. I wrote the following scripts to help me test automatically.

```
# generate segmented input data
for i in `seq 1 10`; do
    echo "Generating segmented input data for data $i"
    perl separator_big5.pl ./test_data/$i.txt > ./test_data/$i.seg
done

# generate all the answers for the homework
mkdir ./ans
for i in `seq 1 10`; do
    echo "Generating answer for data $i"
    disambig -text ./test_data/$i.seg -map ./ZhuYin-Big5.map -lm ./bigram.lm -order 2 > ./ans/$i.txt
done

echo "make and compile the files"
make
# run mydisambig on the test data
mkdir ./result
for i in `seq 1 10`; do
    echo "Generating answer for data $i"
    ./mydisambig -text ./test_data/$i.seg -map ./ZhuYin-Big5.map -lm ./bigram.lm -order 2 > ./result/$i.txt
done

```

```
# Compare the output of mydisambig with the output of disambig
for i in `seq 1 10`; do
    echo "Comparing result for data $i"
    diff ./ans/$i.txt ./result/$i.txt
done

make clean
```

Program

mapping

I use python3 code to generate the mapping from ZhuYin to the possible chinese characters.

mydisambig

Parse the arguement the same as `disambig`. Then read the mapping, language model, input text. Generate the input text based on the language model from the corpus, using viterbi algorithm to find the optimal solution. Finally, output each sentence to stdout for each line. I've written a script to test on all input_data, and then compare them with the results of `disambig`. They output the exact same prediction.