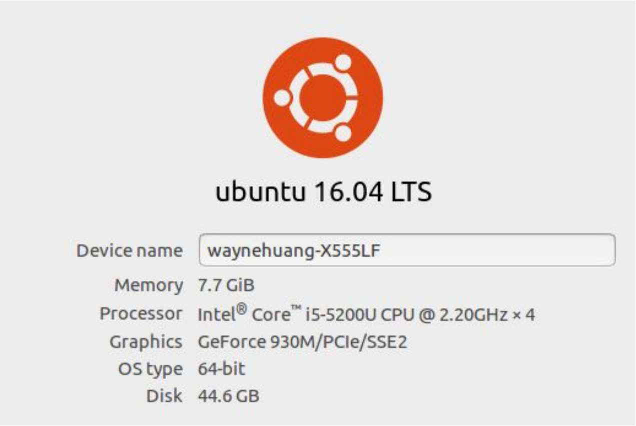
Digital Speech Processing Hw#3

B04501095 黃平瑋

Environment:

OS: Ubuntu 16.04

Machine type:i686-m64



ubuntu 16.04 LTS

Device name waynehuang-X555LF

Memory 7.7 GiB

Processor Intel® Core™ i5-5200U CPU @ 2.20GHz × 4

Graphics GeForce 930M/PCIe/SSE2

OS type 64-bit

Disk 44.6 GB

How to Compile?

$ make

How to Execute?

1. clean files

$ make clean

2. generate mapping

$ make map

3. output result files

$ make run

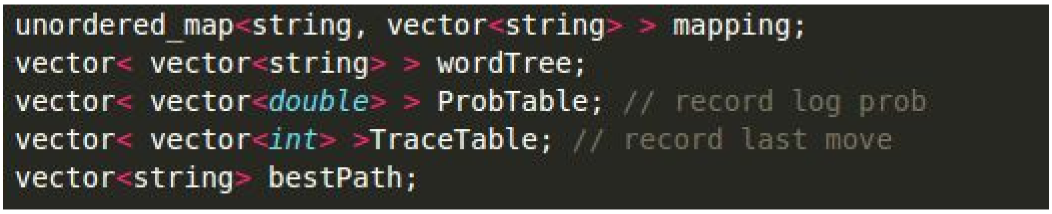
Summary of Program:

1. mapping.py

I construct a dictinary in python to handle the mapping rules. The key is either ZhuYin(注音) or a single word, and the value is a set containing possible Chinese words.

2. mydisambig.cpp

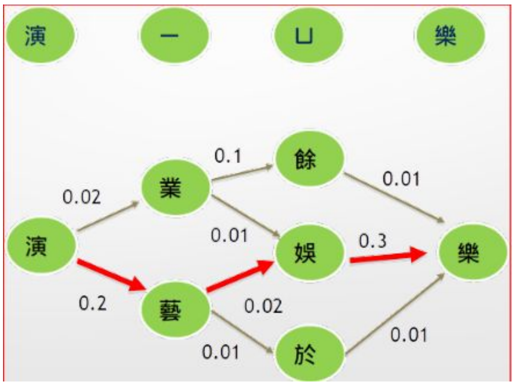
the data member in my class



```
unordered_map<string, vector<string> > mapping;
vector< vector<string> > wordTree;
vector< vector<double> > ProbTable; // record log prob
vector< vector<int> >TraceTable; // record last move
vector<string> bestPath;
```

First, I use a hashmap **mapping** to store the mapping form 'ZhuYin' to Chinese words, the candidate Chinese words are stored in a vector.

Second, I parse the testdata text to a 2-dimensional vector **wordTree,** one dimension is the sequence in text, and the other dimension is the possible mapping. The picture below describe the use of vector.



Then, utilizing language model to calculate bigram probability **log P(w2 | w1)**, and use Viterbi Algorithm to find the best path, and the log probability of each node in **wordTree** is store in **probTable.**

Furthermore, I use a 2-dimensional vector to record the previous node of every node. When viterbi iteration is done, I can find the best path by finding the previous node of every node in the path.