Homework 3

309553048_陳柏丞

- **Execution：**
  - Ubuntu 18.04.5 LTS
    - ./viterbi <input file> <output file>

- **Abstract：**
  - Main：
    - Init
    - Viterbi with log
    - Traceback
    - Calculate accuracy

- **Implementation：**

Init

```
#include <bits/stdc++.h>
#define all(x) x.begin(), x.end()

int main(int argc, char* argv[])
{
    if (argc < 2) return 0;
    std::string input_file = argv[1];
    std::string output_file = argv[2];
    std::ifstream input(input_file);
    std::ofstream output(output_file);

    // init
    std::vector<std::string> type = {"sunny", "foggy", "rainy"};
    std::vector<double> start = {std::log(0.5), std::log(0.25), std::log(0.25)};
    std::vector<std::vector<double>> trans = {
        {std::log(0.8), std::log(0.15), std::log(0.05)},
        {std::log(0.2), std::log(0.5) , std::log(0.3) },
        {std::log(0.2), std::log(0.2) , std::log(0.6) }
    };
    std::vector<std::vector<double>> emis = {
        {std::log(0.9), std::log(0.1)},
        {std::log(0.7), std::log(0.3)},
        {std::log(0.2), std::log(0.8)}
    };
    int T, N = start.size();
    std::vector<int> states, obs;
    std::string tmp;
    input >> T;
    while(input >> tmp)
    {
        auto it = std::find(all(type), tmp.substr(0, tmp.find(',')));
        int state = std::distance(type.begin(), it);
        states.push_back(state);
        int ob = tmp.substr(tmp.find(',')+1) == "yes" ? 1 : 0;
        obs.push_back(ob);
    }
    input.close();
```

Viterbi with log

```
// viterbi with log
    std::vector<std::vector<int>> path(T, std::vector<int> (N, 0));
    std::vector<std::vector<double>> v(T, std::vector<double> (N));
    for(int i=0; i<T; i++)
    {
        for(int j=0; j<N; j++)
        {
            if(i==0)
                v[i][j] = start[j] + emis[j][obs[i]];
            else
            {
                double p = -10e9;
                for(int k=0; k<N; k++)
                {
                    double w = v[i-1][k] + trans[k][j] + emis[j][obs[i]];
                    if (w >= p) p = w, path[i][j] = k;
                }
                v[i][j] = p;
            }
        }
    }
```

Traceback

```
// traceback
    auto it = std::max_element(all(path[T-1]));
    int x = std::distance(path[T-1].begin(), it);
    std::vector<int> ans;
    for(int i=T-1; i>=0; i--)
    {
        ans.push_back(x);
        x = path[i][x];
    }
    std::reverse(all(ans));
```

Calculate accuracy

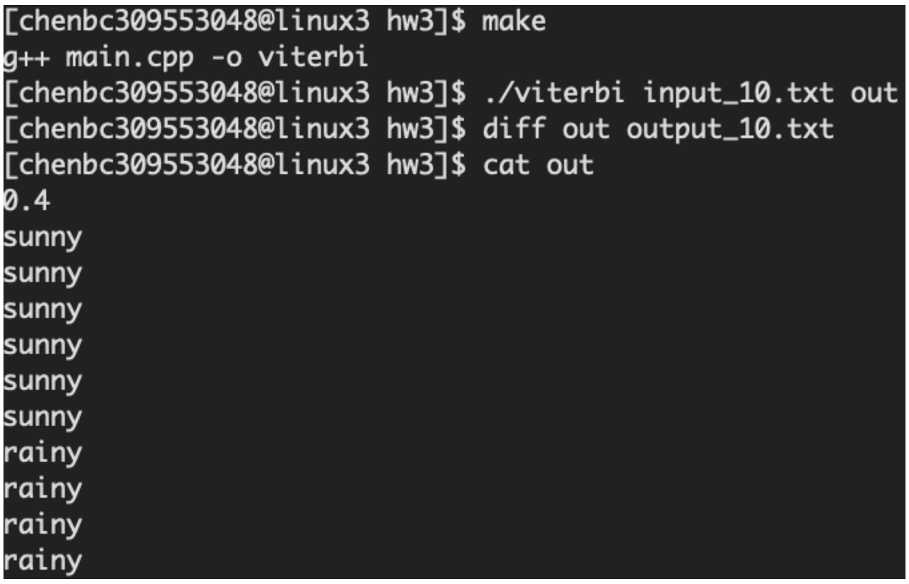
```
// calculate accuracy
    int corr = 0;
    for(int i=0; i<ans.size(); i++)
        if(states[i] == ans[i]) corr++;
    output << (float)corr/T << "\n";

    for(auto e:ans)
        output << type[e] << "\n";
    output << "\n";
    output.close();

    return 0;
}
```

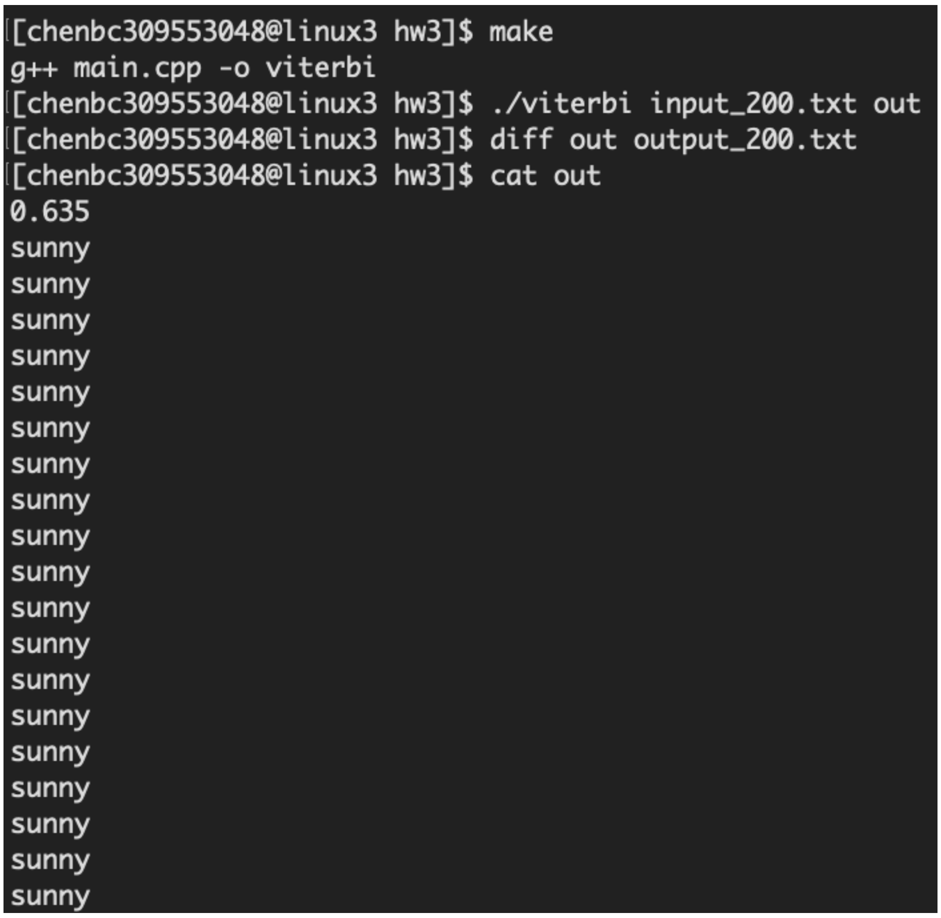
- **Result：**

1. input_10.txt



```
[chenbc309553048@linux3 hw3]$ make
g++ main.cpp -o viterbi
[chenbc309553048@linux3 hw3]$ ./viterbi input_10.txt out
[chenbc309553048@linux3 hw3]$ diff out output_10.txt
[chenbc309553048@linux3 hw3]$ cat out
0.4
sunny
sunny
sunny
sunny
sunny
sunny
rainy
rainy
rainy
rainy
```

2. input_200.txt



```
[chenbc309553048@linux3 hw3]$ make
g++ main.cpp -o viterbi
[chenbc309553048@linux3 hw3]$ ./viterbi input_200.txt out
[chenbc309553048@linux3 hw3]$ diff out output_200.txt
[chenbc309553048@linux3 hw3]$ cat out
0.635
sunny
sunny
sunny
sunny
sunny
sunny
sunny
sunny
sunny
sunny
sunny
sunny
sunny
sunny
sunny
sunny
sunny
sunny
sunny
```