50.007 Machine Learning
Design Project Final Report

Cheng Liying 1000427
Li Nayu 1001912
Wu Hao 1001915

Objective

In this course-related design project, we were aimed at designing the sequence labelling model for informal texts using HMM. We believe that our sequence labelling system for informal texts can serve as the very first step towards building a more complex, intelligent natural language understanding system for mobile texts.

Our python script met all the requirements for Part 2 to 5. This report contains the instructions on how to run our code and the results of each part, as well as the illustration on each part, especially part5, the challenge part.

Output Overview

Part 2:

```
The accuracy score for POS is:
0.600914205345
The accuracy score for NPC is:
0.700003022335
```

Part 3:

```
The accuracy score for POS is:
0.674050632911
The accuracy score for NPC is:
0.776407652552
```

Part 4:

```
The accuracy score of the 10th-best POS tag sequence is:
0.667369901547
```

Part 5:

```
The accuracy score of the new system for POS is:
0.822433192686
```

Instructions on Running the Code

The submitted zip file ‘ML Project’ contains two folders: POS and NPC. The two folders contain all the training data, testing data, outputs and codes respectively.

For checking the results of POS file, you could simply open the ***POS.py*** under the *POS* folder, uncomment the relevant parts of the testing code at the bottom, and run it. **NPC.py** could be checked in the same way.

The outputs are saved in respective files inside each folder.

Implementation Details

- **Part 2**

We simply adopted the approach described in the project document. The results are as following:

```
The accuracy score for POS is:
0.600914205345

The accuracy score for NPC is:
0.700003022335
```

- **Part 3**

Firstly, we used the approach described in the project document. Then, by considering the underflow problem, we took log of both emission and transition parameters. The results are as following:

```
The accuracy score for POS is:
0.674050632911

The accuracy score for NPC is:
0.776407652552
```

- **Part 4**

The basic idea of our algorithm is: for each $\pi(k,u)$, it stored the scores of top-10 best POS tag sequence from state ‘START’ to state ‘u’ at position k instead of only the best score of Viterbi algorithm. In other words, we computed the top-10 best paths for every tag at each position, which finally led to 10 best overall paths. Then we did back tracking to find the top-10 best tag sequence, and calculated the accuracy of the 10th-best tag sequence. The result is shown as following:

```
The accuracy score of the 10th-best POS tag sequence is:
0.667369901547
```

- **Part 5**

In order to improve POS tagger with a better design, we modified several functions in our code by applying other techniques.

1) For emission parameters, we used a smoothing technique called **Absolute Discounting** and modified a bit. The basic formula is as following:

$$e\left(\frac{x}{y}\right)_{new} = \begin{cases} e\left(\frac{x}{y}\right)_{old} - p & for\ existing\ x, and\ e\left(\frac{x}{y}\right)_{old} > 0 \\ \frac{vp}{N} & for\ new\ x \end{cases}$$

Here,

v is number of different words x assigned to a given tag y

N is the total number of different words

P is the penalty for each existing emission parameter, where $p = \frac{1}{T_S+v}$

Ts is the number of existing times of the respective tag y.

2) For transition parameters, we applied **Absolute Discounting** method as well, which is very similar to the one for emission parameters.

3) Apart from this, we did some **pre-processing** on the raw data for those words following very obvious patterns before using HMM, and did **final check** after generating tags by Viterbi Algorithm. This manual and naive method truly improved our HMM accuracy to a certain degree. For example:

“http*”→http

“@*”→@user

“#*”→#HT

……

The results are as following:

```
Total number of predicted tags: 2844
Total number of correctly predicted tags: 2339
The accuracy score of the new system for POS is:
0.822433192686
```