Project2 Report

Rui Chen

March 5, 2020

Additional Feature

In this project, I have implemented the basic operation: **bigram hidden markov model**. In addition, I also implemented **trigram hidden markov model**.

The main difference is that current state is dependent on the previous two states: $p(s_i|s_{i-1}, s_{i-2})$. But the result is much less than the basic bigram operation.

Code Structure

All the source codes are in the file **src**, and all datasets are in the file **data**. There are three code files:

- run.py The main function, and the specific data files are specified in this file.
- bigram_hmm.py The bigram hidden markov model are defined in this file. A class is defined in this file, and the function *predict* will predict the POS of each sentence given the emission and transition matrix.
- utils.py The helper functions are defined in this file.

How to run code

The prediction result ***POS_test.pos*** is in file **data**. To train the bigram_HMM model, run the following line,

python run.py

To get the comparison score, please run

python scorer.py ../data/POS_test.pos ground_truth.pos

Score on Development Corpus

I have tried both **bigram hidden markov model** and **trigram hidden markov model**. The scores are as follows,

bigram

- Train on **POS_dev.pos**. The accuracy is : **93.94**.
- Train on **POS_dev.pos** and **POS_train.pos**. The accuracy is : **96.34**.

trigram

- Train on **POS_dev.pos**. The accuracy is : **84.59**.
- Train on **POS_dev.pos** and **POS_train.pos**. The accuracy is : **90.15**.

From the accuracy, we can see that **trigram** does not necessarily outperform **bigram**, and the accuracy of **bigram** is quite descent.