
Homework 3: Chinese Event Extraction

Yanjian Zhang
16300200020@fudan.edu.cn

Abstract

In this project, I use the HMM and CRF to implement the sequence labeling models for Chinese event extraction. The HMM reached 81.89% accuracy in trigger and 61.11% in argument, while the CRF reached the 94.47% accuracy in trigger and 69.64% in argument.

1 Environment

HMM: Windows 10, python 3.7.1, numpy 1.15.4
CRF: Ubuntu 18.04.2 LTS(Windows Subsystem for Linux), CRF++-0.58

2 HMM

2.1 Step

- Initialize the transition, emission and starting state matrix with the appearing frequency.
- Calculate the probability of transition, emission and starting state.
- Use Viterbi algorithm to predict the labels

2.2 Code Detail

- We use add-lambda smoothing for the calculation of probability of transition, emission and starting state. The lambda we use is 0.01

2.3 Result

```
====trigger labeling result====  
type_correct: 0.8325  
accuracy: 0.8189  
precision: 0.2998  
recall: 0.8453  
F1: 0.4427  
====argument labeling result====  
type_correct: 0.1964  
accuracy: 0.6111  
precision: 0.5086  
recall: 0.906  
F1: 0.6515
```

3 CRF

3.1 Step

- Download the CRF tool from <https://taku910.github.io/crfpp/#download>
- Design the template for features
- Get the result with the following shell command

```
crf_learn template argument_train.txt argument_model
crf_test -m argument_model argument_test.txt > argument_result.txt
crf_learn template trigger_train.txt trigger_model
crf_test -m trigger_model trigger_test.txt > trigger_result.txt
```

3.2 Result

```
====trigger labeling result====
type_correct: 0.9756
accuracy: 0.9447
precision: 0.9762
recall: 0.3591
F1: 0.5251
====argument labeling result====
type_correct: 0.56
accuracy: 0.6964
precision: 0.8099
recall: 0.3177
F1: 0.4564
```

4 Reference

- Application of CRF model in NLP:
<https://blog.csdn.net/oTengYue/article/details/78005557>
- Speech and Language Processing, Chapter 8:
<https://web.stanford.edu/~jurafsky/slp3/8.pdf>