INF 558 Assignment - CRF

Dependencies / Requirements:

1) python 2.7
2) numpy
3) nltk
4) pycrfsuite
5)BeautifulSoup

Commands:

1) To execute **extract.py,** cd into Nikhit_Mago_hw4 and run:

python2 extract.py ucla.model test-ucla.txt test-ucla-markup,txt

The above command will create a file called test-ucla-markup.txt, which will contain the markup text.

2) To execute **train.py**, cd into /train inside Nikhit_Mago_hw4

python2 train.py

The above command will create ucla.model inside the given folder and will print out the train and test performance results. The test set manually created has been fed to this model for performance.

Feature Engineering for CRF:

To create the extraction program, I have used BeautifulSoup to separate the X from Y to train the CRF model. I have used many features for the CRF model. The features are listed below:

1) The word in lowercase
2) A Boolean variable to check if word is uppercase
3) Part-of-Speech Tag of the word
4) The next word in lowercase
5) A Boolean variable to check if next word is uppercase
6) Part-of-Speech Tag of the next word
7) The next to next word in lowercase
8) A Boolean variable to check if next to next word is uppercase
9) Part-of-Speech Tag of the next to next word
10) The previous word in lowercase
11) A Boolean variable to check if previous word is uppercase
12) Part-of-Speech Tag of the previous word
13) The previous to previous word in lowercase
14) A Boolean variable to check if previous to previous word is uppercase
15) Part-of-Speech Tag of the previous to previous word

Basically, I have used +- 2 words for every word in the file to see how neighbouring

words enhance performance. In addition to these features, I have also not separated commas and semi colons from words because they have some significance in predicting tags such as format. For example, I have used “Seminar,” instead of “Seminar”. However, I have used the POS tag of the word “Seminar”, not “Seminar,”.

Train Data Performance Report

```
Tag -> format
F1 score = 1.0
Recall score = 1.0
Precision score = 1.0


Tag -> requisite
F1 score = 0.98
Recall score = 0.99
Precision score = 0.98


Tag -> description
F1 score = 0.97
Recall score = 0.94
Precision score = 1.0


Tag -> grading
F1 score = 0.99
Recall score = 0.99
Precision score = 0.99


Tag -> others
F1 score = 0.97
Recall score = 0.96
Precision score = 0.98
```

Test Data Performance Report

```
Tag -> format
F1 score = 0.99
Recall score = 1.0
Precision score = 0.98
```

```
Tag -> requisite
F1 score = 0.89
Recall score = 0.94
Precision score = 0.85


Tag -> description
F1 score = 0.96
Recall score = 0.94
Precision score = 0.98


Tag -> grading
F1 score = 1.0
Recall score = 1.0
Precision score = 1.0


Tag -> others
F1 score = 0.96
Recall score = 1.0
Precision score = 0.93
```