CS134 (Fall 2012): Programming Assignment 2
Hidden Markov Model

Chuan

October 13, 2012

In this assignment, I think the biggest achievement is recognizing how powerful numpy it is! and I am no longer afraid of computing matrix.

HMM classification results with different features

Test result by separately using (1)word, (2)tag and (3)word& tag as features

Figure 1: word as features

```
row = predicted, column = truth
  B       I       O
B 11116.0 852.0   897.0
I 503.0   12640.0 682.0
O 716.0   965.0   19006.0

B       precision 0.864050      recall 0.901176  F1 0.882222
I       precision 0.914286      recall 0.874317  F1 0.893855
O       precision 0.918741      recall 0.923294  F1 0.921012
accuracy rate = 0.902590
```

Figure 2: tag as features

```
row = predicted, column = truth
  B       I       O
B 11602.0 820.0   530.0
I 318.0   12941.0 387.0
O 415.0   696.0   19668.0

B       precision 0.895769      recall 0.940576  F1 0.917626
I       precision 0.948337      recall 0.895137  F1 0.920969
O       precision 0.946533      recall 0.955453  F1 0.950972
accuracy rate = 0.933174
```

Figure 3: word and tag as features

```
row = predicted, column = truth
  B       I       O
B 11720.0 778.0   456.0
I 281.0   13027.0 318.0
O 334.0   652.0   19811.0

B       precision 0.904740      recall 0.950142  F1 0.926885
I       precision 0.956040      recall 0.901086  F1 0.927750
O       precision 0.952589      recall 0.962400  F1 0.957469
accuracy rate = 0.940499
```

we can see that by using tag as features, we got even better results than using word as results. So maybe that POS has more gain for the Noun phrase chunking. And also by combining the word feature and tag feature, it makes sense that we get even better results since the words have improved the performance by correcting the wrong classification of the POS.