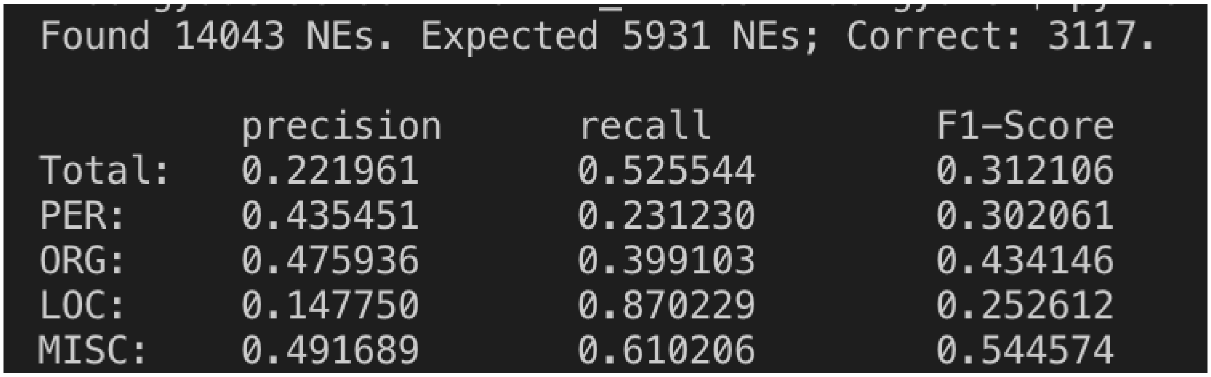
COMS 4705: NLP HW1 Observations

Zhuangyu Ren(zr2209)

February 13, 2019

Question 4

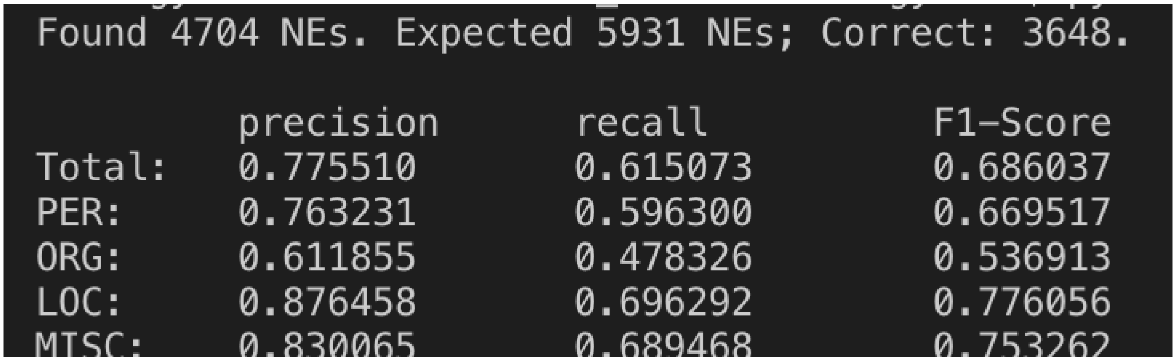


```
Found 14043 NEs. Expected 5931 NEs; Correct: 3117.

          precision      recall         F1-Score
Total:    0.221961       0.525544       0.312106
PER:      0.435451       0.231230       0.302061
ORG:      0.475936       0.399103       0.434146
LOC:      0.147750       0.870229       0.252612
MISC:     0.491689       0.610206       0.544574
```

As we count all the words whose frequency less than 5 as '_RARE_', we assume all these words are of the same type. Acturally, they are not in the same type. So this way of dealing with low-frequency words is not accurate, and the percision rate is very low.

Question 5



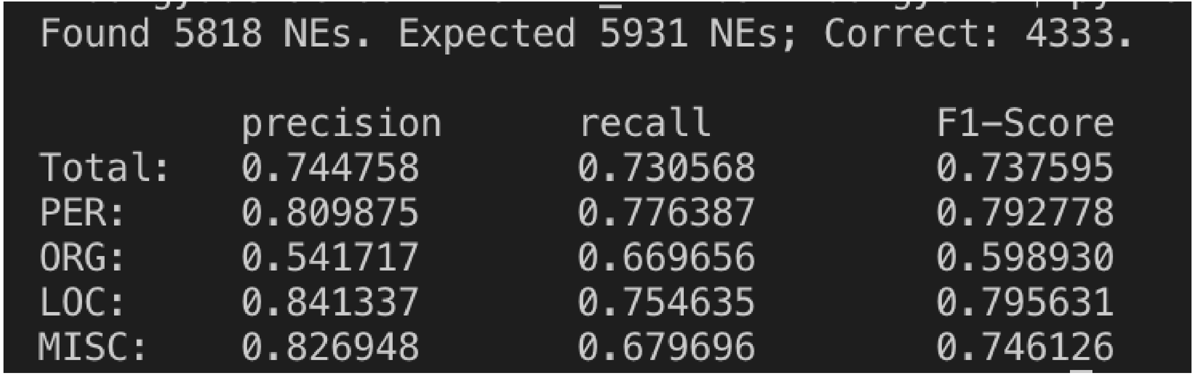
```
Found 4704 NEs. Expected 5931 NEs; Correct: 3648.

          precision      recall         F1-Score
Total:    0.775510       0.615073       0.686037
PER:      0.763231       0.596300       0.669517
ORG:      0.611855       0.478326       0.536913
LOC:      0.876458       0.696292       0.776056
MISC:     0.830065       0.689468       0.753262
```

Using the Viterbi algorithm and counting into the context, we can see a big improvement in the accuracy.

Question 6

I still choose the words according to their frequency.
If the word’s frequency is less than 5, and it is s capitalized word, then I give it a new name ”_PROPER_NAME_”.
If the word’s frequency is less than 5, and it contains only capital letters and dots, then I give them a new name ”_FIRST_NAME_”.
If the word’s frequency is less than 5, and it consists only of numerals, then I give them a new name ”_NUMBER_”.



```
Found 5818 NEs. Expected 5931 NEs; Correct: 4333.

          precision      recall         F1-Score
Total:    0.744758       0.730568       0.737595
PER:      0.809875       0.776387       0.792778
ORG:      0.541717       0.669656       0.598930
LOC:      0.841337       0.754635       0.795631
MISC:     0.826948       0.679696       0.746126
```

Now the accuracy has improved by about 5