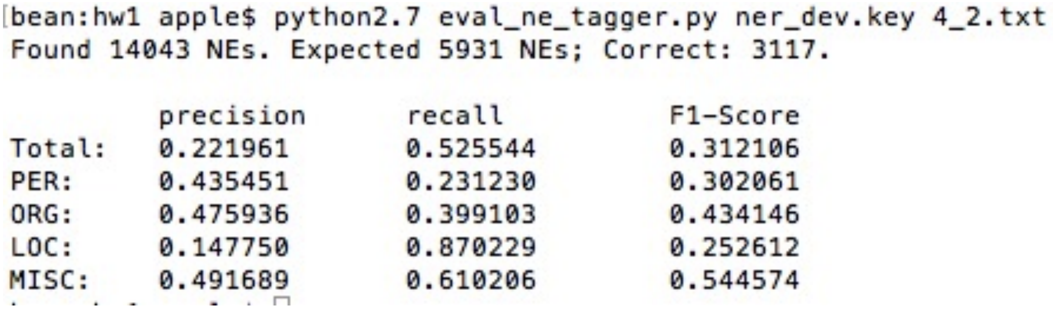
Huibo Zhao
NLP 4705 HW1

Question 4:

./4_1.py # generate ner_train_rare.dat

python2.7 count_freqs.py ner_train_rare.dat > ner_rare.counts

./4_2.py # generate 4_2.txt



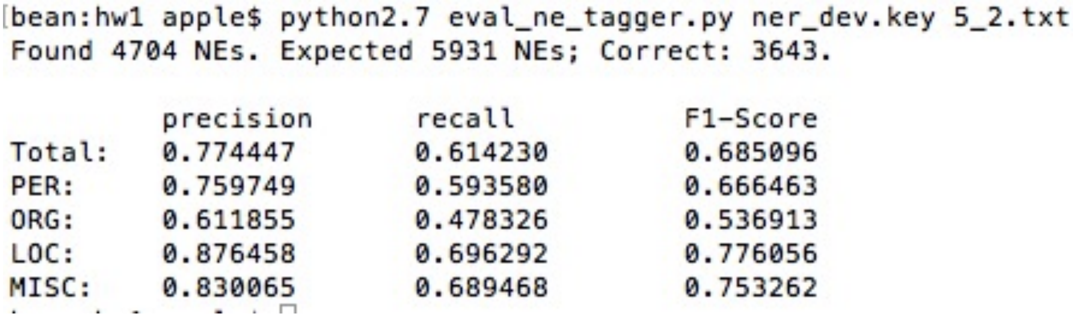
```
bean:hw1 apple$ python2.7 eval_ne_tagger.py ner_dev.key 4_2.txt
Found 14043 NEs. Expected 5931 NEs; Correct: 3117.

         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
```

The performance is not very good, which is in our expectation because it doesn't consider any context. However, the recall score for LOC is very high.

Question 5:

./5_1.py # generae 5_1.txt

./5_2.py # generate 5_2.txt



```
bean:hw1 apple$ python2.7 eval_ne_tagger.py ner_dev.key 5_2.txt
Found 4704 NEs. Expected 5931 NEs; Correct: 3643.

         precision      recall          F1-Score
Total:   0.774447       0.614230        0.685096
PER:     0.759749       0.593580        0.666463
ORG:     0.611855       0.478326        0.536913
LOC:     0.876458       0.696292        0.776056
MISC:    0.830065       0.689468        0.753262
```

Compared with previous result, it improves a lot. It seems that it improves more on precision than recall.

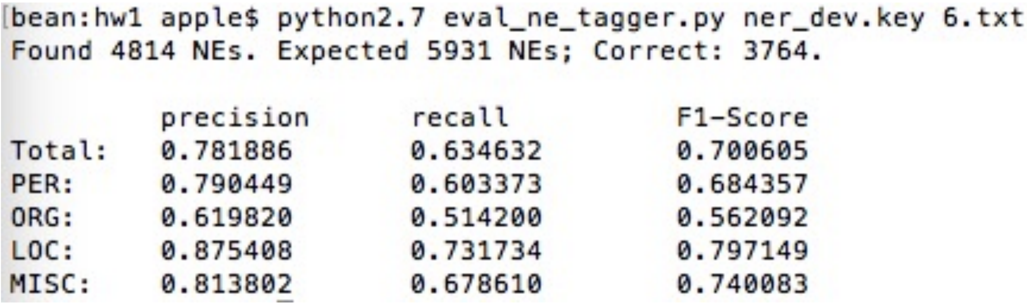
Question 6:

./6_replace.py # generate ner_train_rare2.dat

python2.7 count_freqs.py ner_train_rare2.dat > ner_rare2.counts

./6.py # generate 6.txt

# Instead of grouping rare or unseen word into one category "_RARE", I group them into four categories. They are all capitalized letters with a dot at the end as "_ABR_", all capitalized letters as "_CAP_", all numerals as "_NUM_" and other words that do not belong to any of the previous three categories as "_RARE_"



```
[bean:hw1 apple$ python2.7 eval_ne_tagger.py ner_dev.key 6.txt
Found 4814 NEs. Expected 5931 NEs; Correct: 3764.

         precision       recall          F1-Score
Total:   0.781886        0.634632        0.700605
PER:     0.790449        0.603373        0.684357
ORG:     0.619820        0.514200        0.562092
LOC:     0.875408        0.731734        0.797149
MISC:    0.813802        0.678610        0.740083
```

It doesn't really improve precision score but it improves recall score on ORG and LOC. Generally, it performs slightly better than previous result.