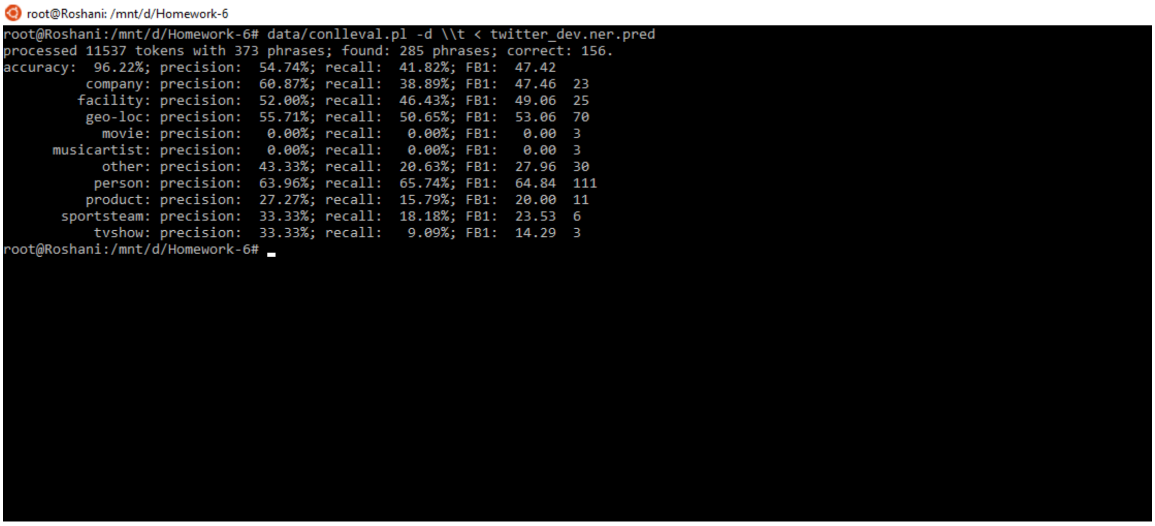
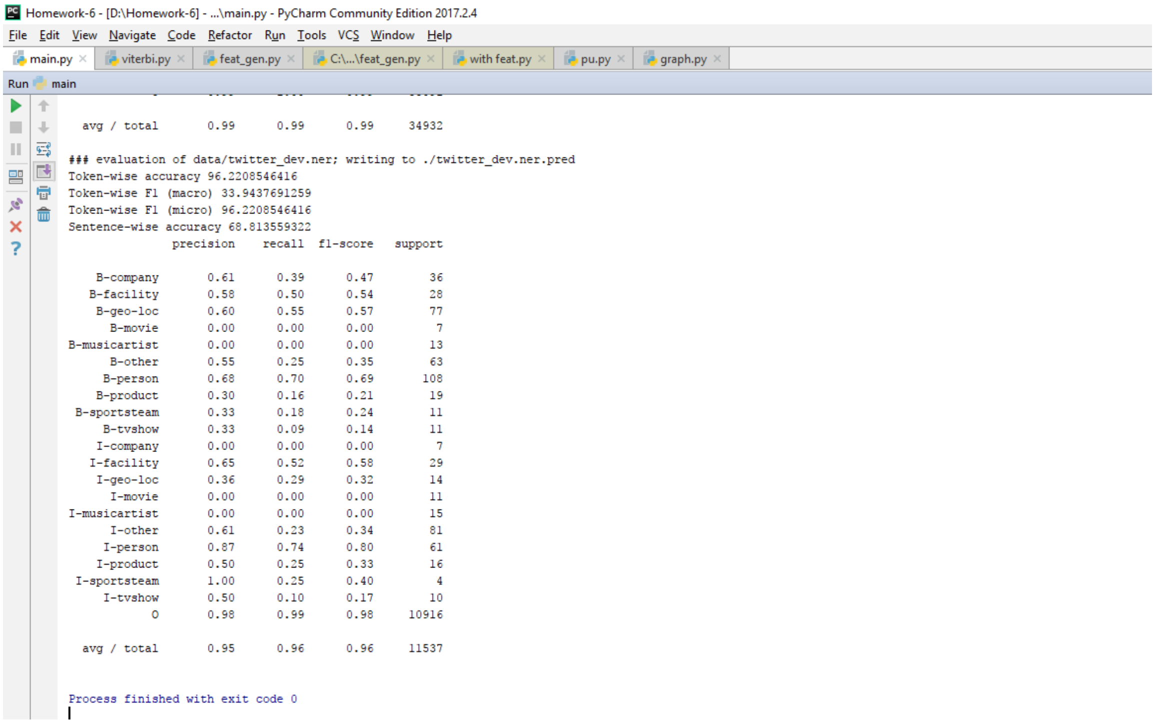
QUESTION:

When you run the experiments, you will and that the evaluation metrics in the provided python script and the CONLL evaluation script are different. Also, the CONLL script reports both accuracy and F1 scores.
Which metrics do you think are the best? Why? When you explain, compare both metrics, and elaborate on your reasons for choosing one over the other.

ANSWER:



```
root@Roshani:/mnt/d/Homework-6# data/conlleval.pl -d \\t < twitter_dev.ner.pred
processed 11537 tokens with 373 phrases; found: 285 phrases; correct: 156.
accuracy:  96.22%; precision:  54.74%; recall:  41.82%; FB1:  47.42
          company: precision:  60.87%; recall:  38.89%; FB1:  47.46  23
         facility: precision:  52.00%; recall:  46.43%; FB1:  49.06  25
          geo-loc: precision:  55.71%; recall:  50.65%; FB1:  53.06  70
            movie: precision:   0.00%; recall:   0.00%; FB1:   0.00  3
      musicartist: precision:   0.00%; recall:   0.00%; FB1:   0.00  3
            other: precision:  43.33%; recall:  20.63%; FB1:  27.96  30
           person: precision:  63.96%; recall:  65.74%; FB1:  64.84  111
          product: precision:  27.27%; recall:  15.79%; FB1:  20.00  11
       sportsteam: precision:  33.33%; recall:  18.18%; FB1:  23.53  6
           tvshow: precision:  33.33%; recall:   9.09%; FB1:  14.29  3
root@Roshani:/mnt/d/Homework-6#
```


Homework-6 - [D:\Homework-6] - ...\main.py - PyCharm Community Edition 2017.2.4
File Edit View Navigate Code Refactor Run Tools VCS Window Help
main.py
viterbi.py
feat_gen.py
C:\...\feat_gen.py
with feat.py
pu.py
graph.py
Run main

```
  avg / total       0.99      0.99      0.99     34932

### evaluation of data/twitter_dev.ner; writing to ./twitter_dev.ner.pred
Token-wise accuracy 96.2208546416
Token-wise F1 (macro) 33.9437691259
Token-wise F1 (micro) 96.2208546416
Sentence-wise accuracy 68.813559322
               precision    recall  f1-score   support

    B-company       0.61      0.39      0.47        36
   B-facility       0.58      0.50      0.54        28
    B-geo-loc       0.60      0.55      0.57        77
      B-movie       0.00      0.00      0.00         7
B-musicartist       0.00      0.00      0.00        13
      B-other       0.55      0.25      0.35        63
     B-person       0.68      0.70      0.69       108
    B-product       0.30      0.16      0.21        19
 B-sportsteam       0.33      0.18      0.24        11
     B-tvshow       0.33      0.09      0.14        11
    I-company       0.00      0.00      0.00         7
   I-facility       0.65      0.52      0.58        29
    I-geo-loc       0.36      0.29      0.32        14
      I-movie       0.00      0.00      0.00        11
I-musicartist       0.00      0.00      0.00        15
      I-other       0.61      0.23      0.34        81
     I-person       0.87      0.74      0.80        61
    I-product       0.50      0.25      0.33        16
 I-sportsteam       1.00      0.25      0.40         4
     I-tvshow       0.50      0.10      0.17        10
            O       0.98      0.99      0.98     10916

  avg / total       0.95      0.96      0.96     11537


Process finished with exit code 0
```

Conlleval.pl evaluation is not dependent on subcategories like intermediate and beginner tags but it only gives the results (that is the accuracy, precision and recall and F1 values) based on the main categories. As the subcategories are not considered, conlleval gives the overall performance evaluation.

On the other hand, the main.py evaluation considers the subcategories while calculating the factors like precision and accuracy etc. So, there are B and I tags that come to consideration.

Therefore, if we want to know the overall performance and not go into details of B I tags we use conlleval.pl and wherever detailed evaluation is needed we use main.py metric.

Also, I would like to mention that the conlleval.pl evaluation metric gives better results as compared to main.py