NLP Assignment1-Report

1.
a. I have split the training data into 80-20 as per the question using random_seed=4. After testing it in my 20% test data macro average of 0.57 was obtained.
b. Classification report:

```
{'For only 80-20 split data (question one)': 0.575}
```

2. False Positive Analysis:
It was observed 'I-Opinion' was falsely predicted if the previous predicted token is 'B-Opinion' irrespective of what a word is. Similarly token "based" was misclassified as 'B-Origin' most of times and token 'classic' was predicted as 'B-Opinion'. Pattern like this have been observed in False Positive with lower precision. It looks like our model is confused as it has no idea what the word actually is whether it is Noun, DT etc. We can improve the precision of these class by feeding more extra feature for training such as previous word or it's POS tag along with it current word POS tag so it becomes easier for our model to decide better and make less misclassification.

3. False Negative Analysis:
BIO tag like 'I-Character_Name' has been classified wrongly as 'I-Plot' which is quite interesting. As one can expect it confusing mostly with 'I-Actor'. It would be have been easier for the model if token POS tag was given as one of the input to training set. Similar issue is observed for BIO tag 'B-Character_Name'. 'I-Soundtrack' (20 tokens) and 'B-Soundtrack' (10 tokens) have been highly misclassified there can be a reason as number of 'I-Soundtrack' and 'B-Soundtrack' are very few in our data set as we haven't provided enough data of these tags for training due to which it has been wrongly predicted.

Conclusion from 3 and 4:

a. We can try different feature combination but mostly we might get high accuracy where with current token and its tag we also add previous words or it's POS tag.
b. If possible provide enough data of all the tags.

4.
Pre-processing and POS tag feature of current token was added in this question.
a. After training with additional feature of POS tag our model showed 0.01 improvement.
As per classification summary:
{'For only 80-20 split data (question one)': 0.575,
'POS Tag on 20% of our test data': 0.5761}

b. While comparing recall from before and after POS tag it is observed tags 'I-Opinion' has further decreased but 'I-Character_Name' and 'B-Character_Name' have increased by 0.5. Where as 'I-So undtrack' and 'B-Soundtrack' remain the same.

c. In the precision comaprision between POS and POS tag. It can be seen 'I-Opinion','B-Origin','I-C haracter_Name precision has decresed by 0.3,0.1 and 0.1 respectively. But for tags 'B-Opinion','B-Plot' precision have increased by 0.3 and 0.1 respectively

Note: I have been looking the same 5 precision and recall classes throughout the report for better understanding and changes can be observed easily.

5. About the data set:
After plotting graph and doing analysis I have come to some conclusion:

1. Median and Average sentence length in our dataset is 19 and 20.28 respectively.
2. We have I-Plot around 38.89% in our train data set.
3. 35.79% are '0' BIO tag meaning there is no special tag for it.
4. Word Starting from B (i.e. B of BIO tag) is 14.6%
5. Word Starting from I (i.e. I of BIO tag is 50.23%
6. Normal words are 35.17%
7. From the graph we can say 'the' word has the maximum number of count 6645 followed by 'a'. Well these are stop words may be in future we can remove stop words before pre-processing.
8. There are no punctuation in our data set. Therefore I have removed adding punctuation feature from get_feature() functions.

While going through False positive and False negative it was observed that our model was quite clueless while predicting some of the token for instance ‘I-Opinion’ was falsely predicted for the token if the previous token prediction was ‘B-Opinion’. Looked like our model needed more input feature other than just some suffix, current word and its POS tag as we did in question 4.

So I have used eight features which mostly involve previous words and its tag. I decided this after my observation for False Negative and False Positive.(I have briefly mentioned it in notebook file)
This features are:

a. Using only one previous word which gave macro average as 0.6174
b. Using only one next word which gave macro average of 0.6034
c. Using only one previous word and one next word which gave macro average of 0.6228
d. Using only two previous words which gave macro average of 0.6245
e. Using only two previous word POS tags which gave macro average as 0.6226
f. Using only two previous word and it’s POS tags macro average as 0.6365
g. Using only one previous word and one next word which gave macro average as 0.6234
h. Using only one previous word and it’s tag which gave macro average of 0.6239

After plotting the graph and from above data it can be clearly said that “two previous word and its POS tags” has the maximum macro average.
After using these features for my training whole dataset and testing actual test data provided I achieved macro average of 0.637 which can be further improved by hyper parameter (as shown in code)

From this feature we can also observe how our actual low precision and low recall classes have performed. (I’m focusing only on those classes which I had got the lowest after normal split)
After running code for precision and recall, for specific this feature as it gave highest macro average.

It is observed BIO tag ‘I-Opinion’, ‘I-Character_Name’ and ‘B-Character_Name’ have increased approximately 50% of its actual recall rate.

For precision class all BIO tags precision has increased except for tags ‘I-Character_Name’ and ‘B-Origin. This can prove that our intuition for feeding more input based on it’s previous, next words or their respective tokens were right.

All the output can be observed clearly in the code.