Erblin Marku, ID:210762815, Msc Computer Science
NLP, ECS763P

1. **Split the training data into 80% training, 20% development set**
   I achieved this task by using the *train_test_split()* from the *scikit-learn* library to divide my dataset into two subsets. The code is in line 10 in the Jupyter Notebook submitted. Since our data is already in paired tuples, we do not need to use the *y_train* and *y_test* right now. Using a *random_state* of a fixed value of integer for the split helps me to see the changes when I try to improve the tagger to get better results, since the split will be the same each time I run the code.
2. ***False Positives*** We have 5 classes, *B-Soundtrack, I-Soundtrack, I-Opinion, B-Quote and I-Quote*, *B-Soundtrack and I-Soundtrack*
   These two classes have the lowest precision (Pre=0.0) with a low number of occurrences in the corpus but none of them correctly predicted. Only are 5 FP mistakes labelled in this categories from the O , B-Director and I-Director classes, all in the same sentences understandable because the words are very popular in the music genre. Weighing in how these words are predictive for each category can be useful
   I-Opinion
   It has a very low precision (Prec= 0.13) with a lot of FP happening in the O class. From the sentences we get that the mistakes happen because a lot of the sentences include opinions for movies and these opinions are expanded with a lot of words forming long opinions, usually making about 3 to 5 words FP in the same sentences. Same solution as above could be helpful
   I-Quote and B-Quote
   Their precision is very low being 0.24 and 0.22 respectively where they are most mistaken for the *I-Plot* and *O* categories. They usually tend to be word that are in a group in the last part of the sentences and instead of being just outside single words, they are taken as a group and predicted to be quotes. Recognition of quotation marks can make it easy, including single quotation mark that happens usually in names.
3. ***False Negatives*** We have 5 classes, *I-Soundtrack, B-Soundtrack , I-Opinion, B-Quote, B-Character_Name B-Soundtrack and I-Soundtrack*
   These two classes have the lowest recall (Rec=0.0) missing all of their predictions. The most of them go to the O class, same explanation as in the FP applies.
   B-Quote
   The recall Is at 0.11 missing most of its predictions to the I-Plot class. It is difficult to undersat the beginning of a quote with the pre-processing done to the words. Recognition of quotation marks can make it easy, including single quotation mark that happens usually in names.
   I-Opinion
   It has slightly lower recall than its precision (Rec= 0.13) missing most of predictions to *O* and *I-Plot*. Long and mixed long structured sentences cause these mistakes. Using bigram models or more complex can be useful.
   B-Character Name
   The class has very low recall regarding its precision (Rec=0.24, Pre=0.76) missing around three fourth of its predictions, most of them going to I-Plot and O classes. Usually, the Character names are inside the plot in the sentences, so the false prediction is perfectly understandable.
4. **POS tagger**
   After adding the POS tagger in our training data, we have successfully improved the Macro average of f1 score by very little (from Macro avg=0.562549 to Macro avg=0.562549). The POS tagger made some of the categories improve their *f1 score* and some others fail a little more in terms of FP and FN. Overall it does good on *B-Quote* and *I-Quote* classes by improving their precision and recall. *B-Quote* gets a 0.14 improvement on the *f1 score*. *I-Opinion* gets better by 0.05 in *f1 score* also. Important is that most of the classes with low precision got a little better and the classes with low recall got predicted more correctly.

5. In this question first I am adding more levels of suffix and we see that we have an improvement of the *Macro Avg* of the *f1 score* by 0.03, which is seen having the most impact on the *B-Character_Name* class by increasing this ones f1 score by 0.11. All our least effective classes see a little improvement, ex. *I-Opinion* is increased by 0.04 and *I-Quote* by 0.05. Many of the mistakes are words that have prefixes and so II added more levels of prefixes also and that helped also by increasing the *Macro avg f1 score* with 0.01. I tried to add more levels of prefixes and suffixes and I went up to 6. My *Macro avg f1 csore* for this was 0.60. All the classification reports and differences are shown in tables in the Excel doc included in the submission.
   Next step, for even better performance I tried optimizing the hyperparameters by adjusting the $training\_opt$ dictionary values. First try was with minimum document frequency set at 200 and *c1* and *c2* parameters set to 0.1 respectively. The *Macro avg* did not change but what I can see from the classification report the scores of the troubled classes got worse,
   I am trying only the Lasso regression to see if it works better since we have many features. So, I set c2=0 and run again, this time my *Macro avg* was 0.61. Some of the classes drop and some go up, but the recall is an improvement in most of them. So, the Lasso regression is getting the FN.
   I will try the Ridge regression even though the Lasso regression should give better result in feature extraction by eradicating the features with weights very close to 0. The classification report shows that Lasso is better for our problem.
   Although giving the same result for the *Macro avg f1 score* we see that Lasso regression performs better on our problematic classes by showing improvement overall in the 5 classes with most FP and NP like *I-Quote and B-Quote, I-Opinion*. The *Macro Avg* continues to stay at 0.61.
   Now I am aiming to get both to work better, so I will increase the c1 and I will set c2 to a small value. I am doing this so I can sacrifice in terms of precision and recal some of the classes with the lowest occurrences in the corpus and get the *Macro avg f1* score up for the model.
   From the results we see that it performs better, the *Macro avg precision* is better than both the previous try's (Macro avg precision = 0.65).
   After playing with the *c1* and *c2* coeficients the best result is at *c1=0.5* and *c2=0.05* where *Macro avg* is at 0.61.
   After running the train process on all our training data and running the test again but with the test subset we did not use before we got a similar Classification report with our best one. The main difference is one increase in *the I-Plot f1 score*, which does not affect the *Macro avg* because of the fluctuation on the other classes.
6. **Different approach**
   To get the f1 score up I needed something more so I implemented another CRF called *CRFsuite* which I found was very flexible to work with our dataset and had a feature function which took into consideration previous and next word, putting them in the features vector. It used the same technique for tagging the training part was more effective since it propagates *Forward/backward algorithm using the scaling method*. Since our sentences produced more FP and FN once one word was predicted wrong , meaning the words that followed or previous were predicted wrong also, this model with the forward/backward algorithm did good. The experiment produced a Macro avg of 0.74, and improvement in most of the categories.
   All classification reports with the difference tables and Confusion Matrixes are found in the Excel doc submitted.