Assignment 1: CRF sequence tagging for Movie Queries

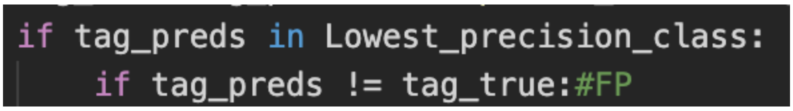
Shuchang Liu-210116524

1. Split the training data into 80% training, 20% development set (5 marks)

   I split the training data into two parts by order: the first 80%(6252 instances) and the last 20%(1564 instances).

2. Error analysis 1: False positives (5 marks)

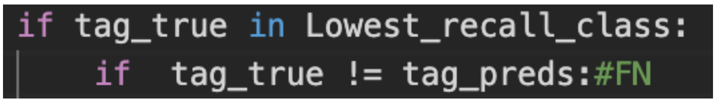
   According to the confusion matrix, in the stand of prediction, the FP code goes like this:



```
if tag_preds in Lowest_precision_class:
    if tag_preds != tag_true:#FP
```

3. Error analysis 2: False negatives (5 marks)

   Similar to task2, but in perspective of ground truth, the FN code goes like this:



```
if tag_true in Lowest_recall_class:
    if  tag_true != tag_preds:#FN
```

   To reduce this error, as well as task 2, I think the adding some of the features into training set can be significant improvement. With the specific tests in task5, my answer to this question is feature prefix.

4. Incorporating POS tags as features (15 marks)

   PreProcess:

   1. Use data to generate related POS tags.
   2. Concatenate tags, word and symbol(“@”) together.
   3. Return the combined data.

   Get the Pos tag:

   1. Make a judgment on the word including symbol(“@”) or Not
   2. If symbol exists, split the word with it.
   3. Add two split result(Word, Pos Tag) into features.

The differences of the features with POS and those without are:

1. With the POS, the word label has become more predictive.
2. With the POS ,The f1-score has slight improvement.

The recorded data for comparison is in the .ipynb file.

5. Feature experimentation and other optimization for optimal macro average (30 marks)

Pre word(N=1)	prefix	Pre tag	Word_POS	Suff tag	Sufflix	Suff word(N)	f-score

N	N	N	Y	N	N	N	0.54
Y	N	N	Y	N	N	N	0.56
N	Y	N	Y	N	N	N	0.63
N	N	Y	Y	N	N	N	0.56
N	N	N	Y	Y	N	N	0.55
N	N	N	Y	N	Y	N	0.55
N	N	N	Y	N	N	Y	0.55
N	N	N	Y	Y	Y	Y	0.56
Y	Y	Y	Y	N	N	N	0.65
Y	Y	Y	Y	Y	Y	Y	0.63

Table 1 feature

Pre word	f-score
N=1	0.6511
N=2	0.6433
N=3	0.6453

Table 2 N-gram

minfreq	f-score
1	0.6511
2	0.6488
3	0.6474

Table 3 MinFreq

The relative features that might influence f-score is listed in Table 1 above. From the table we can see the prefix of the word can improve the result efficiently, While others also have insignificant advantage. The most ideal combination of the features would be: Preword, Prelix, Pretag.

With The best outcome from table 1, we can test the optimal N for N-Gram mode is 1. Similarly, we can get the optimal minifreq is also 1.

Finally, I use current best features to re-train the entire training data (trivia10k13train.bio.txt) and the test data(trivia10k13test.bio.txt) as test set. The final macro avg f-score is 0.62 which is close to our 8-2 training result.

From this assignment, I have completely acknowledged how to test the best model by adding relevant features and become aware of how important the features can be. The technique and patience matter most!