3.1>

CRF vs Logistic Regeression:

CRF is multi class Logistic Regeression model

When you restrict the number of possible outcomes of a CRF model to 2, we get a binary logistic regression model.

And as we know, Logistic Regression tends to normalize locally which leads to label biasing.

$$P(\boldsymbol{t}|\boldsymbol{w}) = \prod_i P(t_i|w_i, t_{i-1}) = \frac{\prod_i \exp(f(t_i,t_{i-1},w_i))}{\prod_i \sum_t \exp(f(t,t_{i-1},w_i))}$$

while CRF tends to normalize globally

$$P(\boldsymbol{t}|\boldsymbol{w}) = \frac{\exp\ (\mathrm{S}(\boldsymbol{t},\boldsymbol{w}))}{\sum_{\boldsymbol{t}} \exp(S(\boldsymbol{t},\boldsymbol{w}))} = \frac{\prod_i \exp(f(t_i,t_{i-1},w_i))}{\sum_{\boldsymbol{t}} \prod_i \exp(f(t_i,t_{i-1},w_i))}$$

In CRF we tend to get the most optimal output by using Inference algorithms like Viterbi and so it gives the best scoring output given the model.Hence its is better than Logistic Regression.

3.2>

BEFORE ADDING FEATURES:

Python script metrics -before adding features			
Logistic Regression			
metric	**twitter_dev.ner.pred**	**twitter_dev_test.ner.pred**	**twitter_test.ner**
Token-wise accuracy	95.5361012395	91.0152104705	98.3735177866
Token-wise F1 (macro)	21.5780375334	10.9195384447	7.08429221834
Token-wise F1 (micro)	95.5361012395	91.0152104705	98.3735177866
Sentence-wise accuracy	66.6101694915	48.6486486486	81.4744200826
Avg F1 score	**0.94**	**0.88**	0.99

Python script metrics -before adding features			
CRF			
metric	**twitter_dev.ner.pred**	**twitter_dev_test.ner.pred**	**twitter_test.ner**
Token-wise accuracy	95.7701308832	91.3070392642	97.0948616601
Token-wise F1 (macro)	29.5648858833	17.9817691763	4.92630101273
Token-wise F1 (micro)	95.7701308832	91.3070392642	97.0948616601
Sentence-wise accuracy	68.6440677966	50.4978662873	76.3902129012
Avg F1 score	**0.95**	**0.89**	0.99

Avg F1 Score for dev and dev-test before adding features for Logistic Regression :
0.94+0.88/2 = 0.91 = A

Avg F1 Score for dev and dev-test before adding features for CRF : 0.95+0.88/2 = 0.915 = B

Based on this, B > A , we can roughly say that CRF gave a higher Avg F1 score(of dev and dev-test) by a difference of B-A = 0.005

Conlleval script metrics -before adding features		
Logistic Regression		
metric	**twitter_dev.ner.pred**	**twitter_dev_test.ner.pred**
Accuracy	95.54%	91.02%
Precision	49.61%	32.35%
Recall	16.89%	8.54%
FB1	**25.20**	**13.51**

Conlleval script metrics -before adding features		
CRF		
metric	**twitter_dev.ner.pred**	**twitter_dev_test.ner.pred**
Accuracy	95.77%	91.31%
Precision	60.61%	46.82%
Recall	26.81%	15.99%
FB1	**37.17**	**23.84**

Avg FB1 Score for dev and dev-test before adding features for Logistic Regression :
25.2+13.51 / 2 = 19.355 = C

Avg FB1 Score for dev and dev-test before adding features for CRF : 37.17+23.84 / 2 = 30.49 = D

Based on this, D > C , we can roughly say that CRF gave a higher Avg F1 score(of dev and dev-test) by a difference of D-C = 11.135

<u>Conclusion:</u>

Based on the average FB1 scores(C and D) from Conlleval metrics and based on the average F1 scores(A and B) from Python script metrics, it is evident that CRF performed better than Logistic Regression before adding features. However, the difference is marginal in case of the Python script metrics(0.005) and it is significant in case of the Conlleval metrics(11.135)

AFTER ADDING FEATURES:

Python script metrics -after adding features			
Logistic Regression			
metric	**twitter_dev.ner.pred**	**twitter_dev_test.ner.pred**	**twitter_test.ner**
Token-wise accuracy	95.8654762937	91.5458082773	96.2924901186
Token-wise F1 (macro)	26.7681222275	16.6265947943	5.45062399599
Token-wise F1 (micro)	95.8654762937	91.5458082773	96.2924901186
Sentence-wise accuracy	67.6271186441	49.5021337127	68.9863361932
Avg F1 score	**0.95**	**0.89**	0.98

Python script metrics -after adding features			
CRF			
metric	**twitter_dev.ner.pred**	**twitter_dev_test.ner.pred**	**twitter_test.ner**
Token-wise accuracy	96.0648348791	91.9260700389	94.9090909091
Token-wise F1 (macro)	31.9997259261	22.111402421	4.63752665245
Token-wise F1 (micro)	96.0648348791	91.9260700389	94.9090909091
Sentence-wise accuracy	69.6610169492	52.0625889047	64.4423260248
Avg F1 score	**0.95**	**0.90**	0.97

Avg F1 Score for dev and dev-test after adding features : 0.95+0.89 / 2 = 0.92 = A

Avg F1 Score for dev and dev-test after adding features: 0.95+0.90 / 2 = 0.925 = B

Based on this, B > A , we can roughly say that CRF gave a higher Avg F1 score(of dev and dev-test) by a difference of B-A = 0.005

Conlleval script metrics -after adding features		
Logistic Regression		
metric	**twitter_dev.ner.pred**	**twitter_dev_test.ner.pred**
Accuracy	95.87%	91.55%
Precision	47.52%	29.64%
Recall	25.74%	15.37%
FB1	**33.39**	**20.25**

Conlleval script metrics -after adding features		
CRF		
metric	**twitter_dev.ner.pred**	**twitter_dev_test.ner.pred**
Accuracy	96.06%	91.93%
Precision	58.74%	43.58%
Recall	16.89%	24.22%
FB1	**43.96**	**31.14**

Avg FB1 Score for dev and dev-test after adding features for Logistic Regression :
33.39+20.25 / 2 = 26.82 = C

Avg FB1 Score for dev and dev-test after adding features for CRF : 43.96+31.14 / 2 = 37.55 = D

Based on this, D > C , we can roughly say that CRF gave a higher Avg F1 score(of dev and dev-test) by a difference of D-C = 10.73

Conclusion:

Based on the average FB1 scores(C and D) from Conlleval metrics and based on the average F1 scores(A and B) from Python script metrics, it is evident that CRF performed better than Logistic Regression after adding features. However, the difference is marginal in case of the Python script metrics(0.005) and it is significant in case of the Conlleval metrics(10.73)

Final Overall Conclusion :

CRF does much better than Logistic Regression in both Python script as well as Conlleval metrics for both the cases before adding features and after adding features.