Named Entity Recognition using BiLSTM and its variants

Harman Singh Shubham Mittal

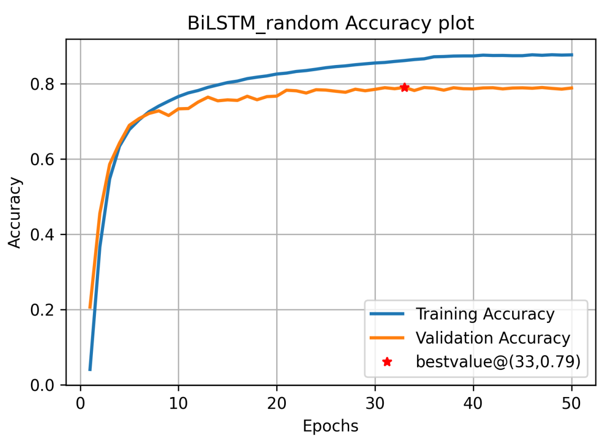
Simple BiLSTM results

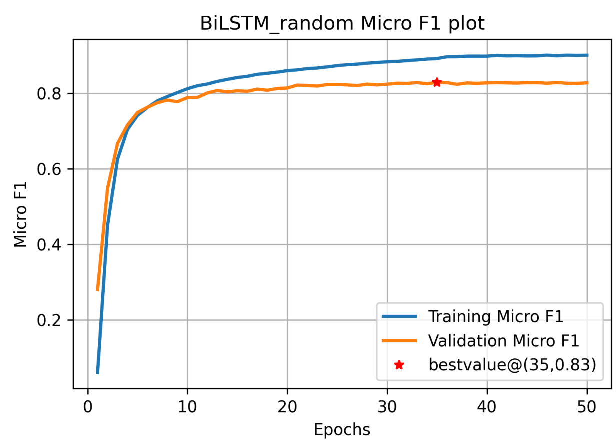
- **BiLSTM with Random Word Embeddings**

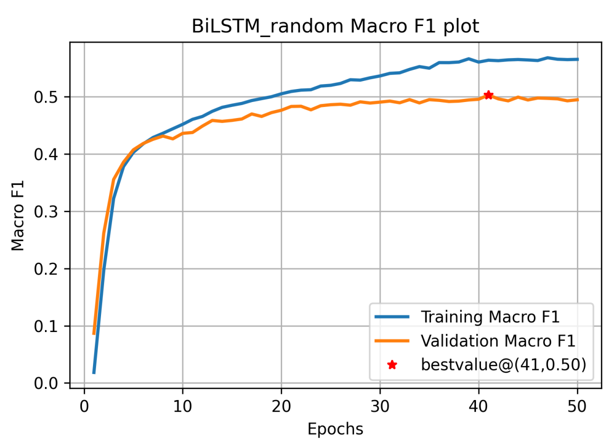
  CONLL based final performance metrics (after training):

	Accuracy	Micro F1	Macro F1
Train	0.982	0.889	0.755
Validation	0.964	0.803	0.619
Test	0.964	0.804	0.614

Plots for performance statistics vs epochs:





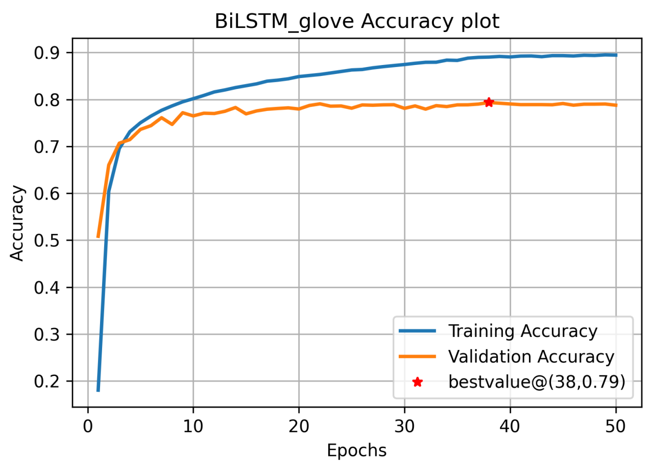


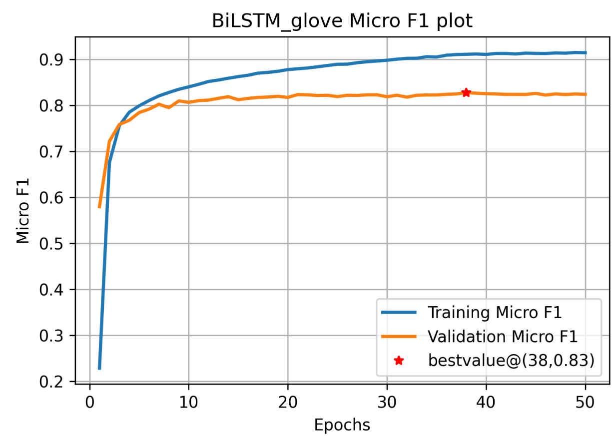
- **BiLSTM with Pretrained (Glove) word embeddings**

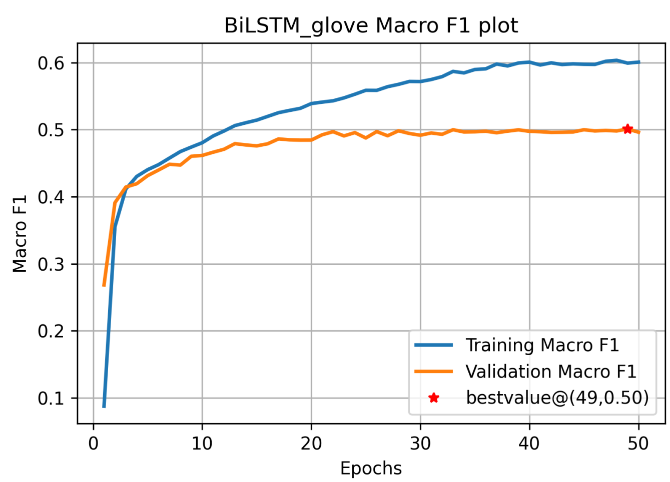
  CONLL based final performance metrics (after training):

	Accuracy	Micro F1	Macro F1
Train	0.986	0.909	0.823
Validation	0.961	0.793	0.620
Test	0.962	0.794	0.614

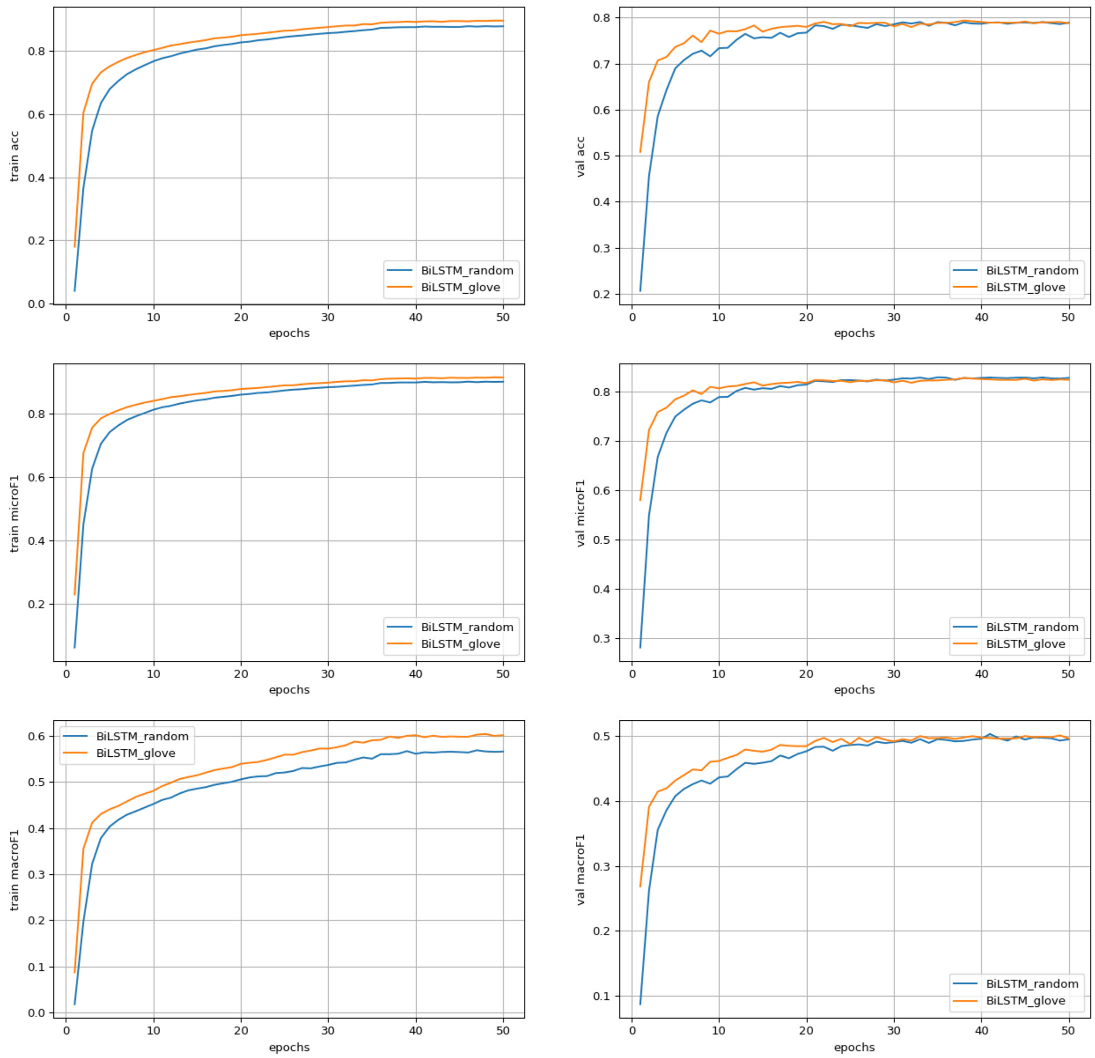
Plots for performance statistics vs epochs:







- **Comparison of Simple BiLSTM with Random and Glove embeddings**
  (Following Plots, 3 rows respectively are for accuracy, microF1 and macroF1. Left column - Training, Right column - Validation)
  (orange line - glove, blue line - random)



Comments

- We see that with glove embeddings the model trains faster than with random embeddings (ie with glove embeddings it takes less number of epochs for same performance). We conclude this because for all the plots, accuracy, microF1 and MacroF1, for both train and validation sets, the curve for BiLSTM with glove embeddings remains above that of random embeddings
- Faster training is expected because glove embeddings provide embeddings, which are similar for similar words, and this pretraining of the embeddings means that we donot have to learn the embedding now, we can just fine tune it during training. While in case of random embeddings all the embeddings also have to be learnt which takes more time.

- The final performance of both BiLSTM's are similar hence we can say that random embeddings are changed during training (learnt) to represent meaningful information from words just like glove embeddings.

Results using BiLSTM using character level features and pretrained embeddings

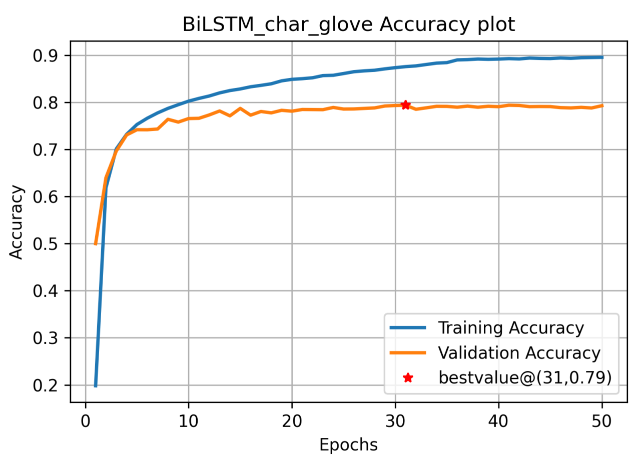
We use a separate BiLSTM for generating forward and backward character level embeddings of the Word which is concatenated with the word embeddings. Total embedding size now becomes 100+25+25 = 150

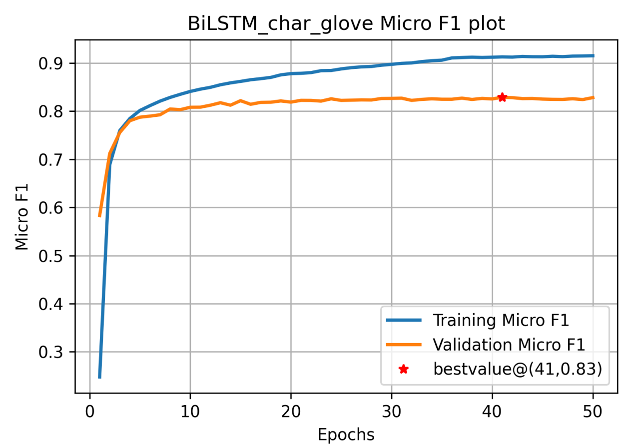
- **BiLSTM with Character Level Embeddings + Pretrained (Glove) word embeddings (concatenated)**

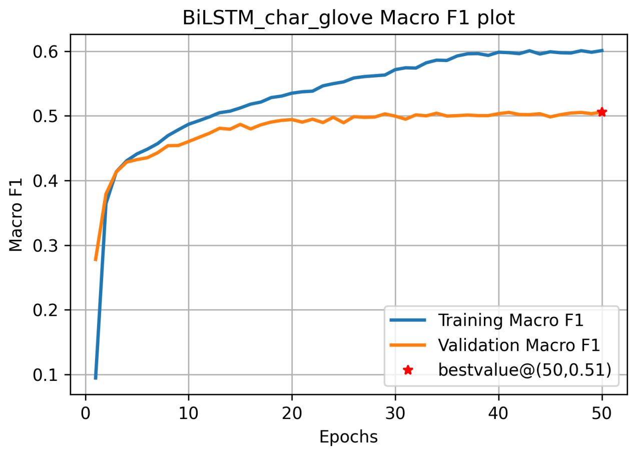
CONLL based final performance metrics (after training):

	Accuracy	Micro F1	Macro F1
Train	0.986	0.908	0.820
Validation	0.961	0.795	0.629
Test	0.962	0.795	0.626

Plots for performance statistics vs epochs:





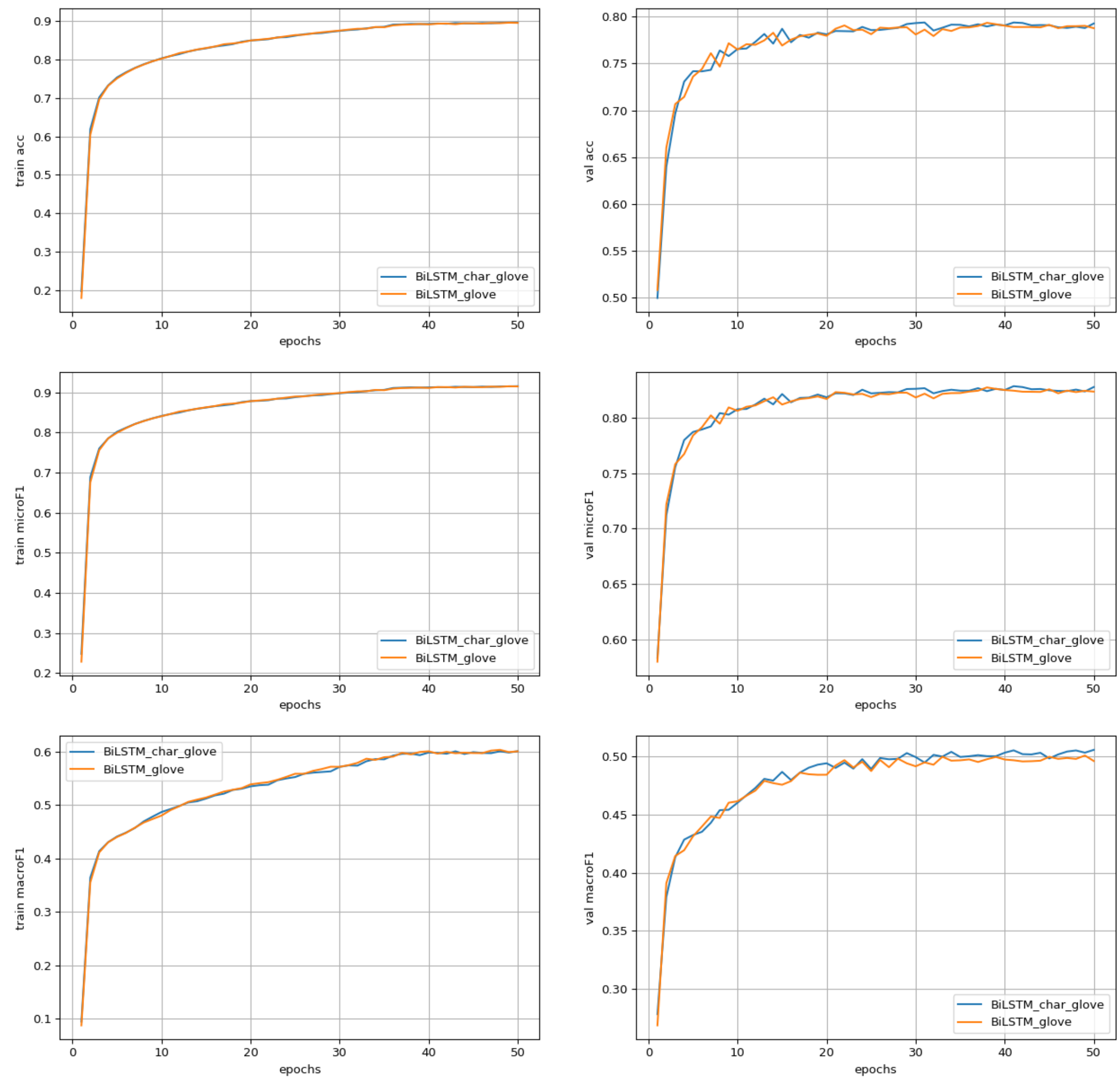


- **Comparison between BiLSTM (Char Level + Glove) model with Bilstm (Glove)**

On the basis of Final Performance:

Model	Validation		Test	
	MicroF1	MacroF1	MicroF1	MacroF1
BiLSTM (Glove embeds)	0.793	0.620	0.794	0.614
BiLSTM (Char Level + Glove embeds)	**0.795**	**0.629**	**0.795**	**0.626**

On the basis of learning curves:
(Following Plots, 3 rows respectively are for accuracy, microF1 and macroF1. Left column - Training, Right column - Validation)
(orange line - glove, blue line - char+glove)



Comments :

- For Validation and Test data, the BiLSTM with character and word embeddings perform better in terms of both micro and macro F1 score. This is expected since we are giving the model character level information of the words, hence this will help enhancing the embeddings of the

words (giving more information) and would also be helpful to words which are not in the training set (out of vocabulary), to have better representation than the standard “<OOV>” tag during validation and test time.
- The training curves for both models are very similar, hence speed of learning for both the models is almost the same, ie the additional lstm required for getting the character level embeddings does not consume more time (this is due to fully vectorized implementation to get char level embeddings without looping over words).

Results using BiLSTM with LayerNormalization, using character level features and pretrained embeddings

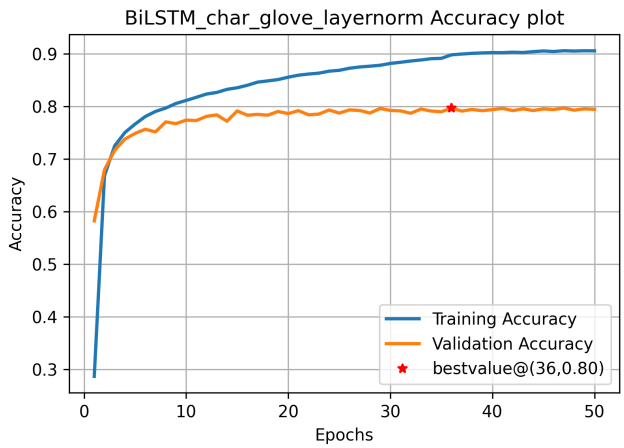
- **Implementation details of LSTM with Layer Normalization:**
  1. We replace nn.LSTM with our implementation of LSTM, which uses layer-normed LSTM Cell implementation.
  2. Inside the LSTM cell, since there are 5 non-linearities, thus the input to them is normalized.
  3. In total, three layer norm layers are used. Two are applied on the linear transformations : $W_{x2h}.x_t$ and $W_{h2h}.h_{t-1}$. Third layer norm layer is applied on c_t .

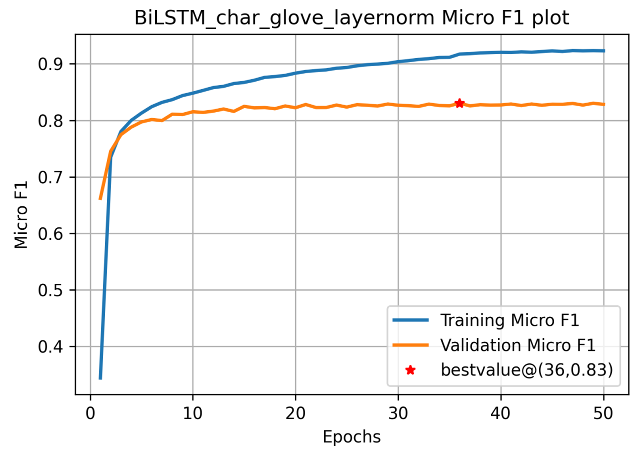
- **BiLSTM with Layer Normalization (used Character Level Embeddings + Pretrained (Glove) word embeddings (concatenated))**

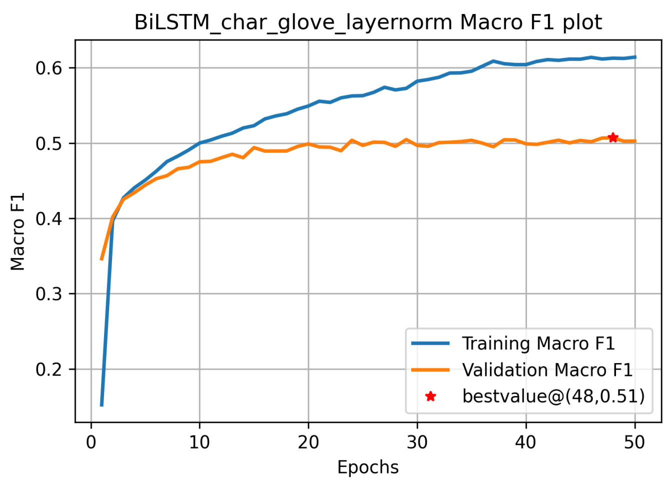
  CONLL based final performance metrics (after training):

	Accuracy	Micro F1	Macro F1
Train	0.989	0.923	0.851
Validation	0.963	0.799	0.625
Test	0.963	0.802	0.627

Plots for performance statistics vs epochs:







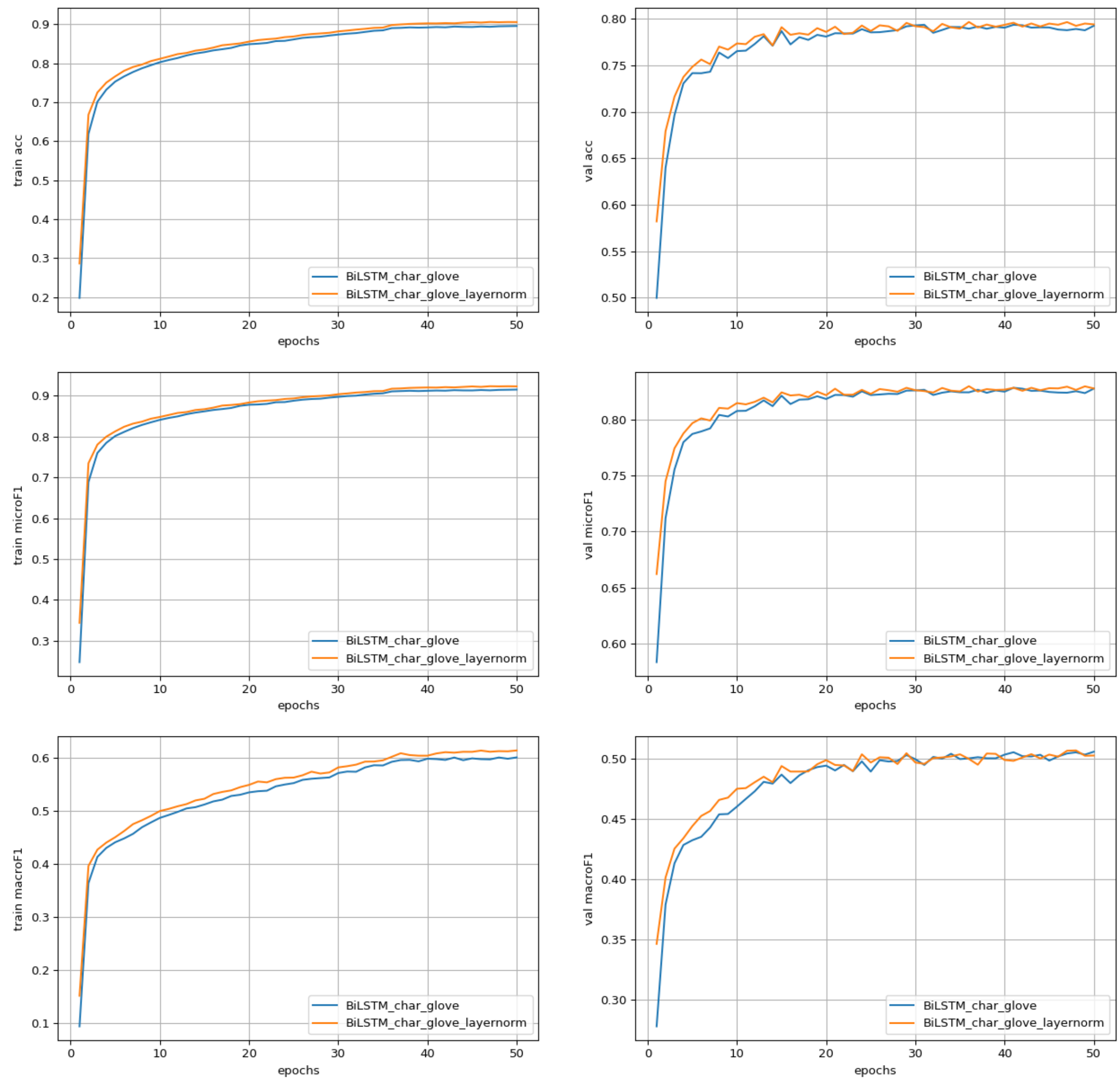
- **Comparison between BiLSTM (Char Level + Glove) with and without Layer Normalization**

On the basis of Final Performance:

Model	Validation		Test	
	MicroF1	MacroF1	MicroF1	MacroF1
BiLSTM (Glove embeds)	0.793	0.620	0.794	0.614
BiLSTM (Char Level + Glove embeds) Without Layer Norm	0.795	0.629	0.795	0.626

BiLSTM (Char Level + Glove embeds) **With Layer Norm**	**0.799**	**0.625**	**0.802**	**0.627**

On the basis of learning curves:
(Following Plots, 3 rows respectively are for accuracy, microF1 and macroF1. Left column - Training, Right column - Validation)
(orange line - char+glove+layernorm, blue line - char+glove)



Comments :

- For Validation and Test data, the BiLSTM with Layer Normalization performs better in terms of Micro F1 score and similar in terms of macro F1 score on test data.
- In most of the curves above, normalized variant of the model, has its curve above the unnormalized. Hence speed of training with layer normalization is more that without it. This is expected because loss landscape with normalization becomes smoother and even (in terms of

gradient) in all directions which lets the model take larger gradient steps (in case of adam optimizer which we have used) hence faster training.

Linear chain CRF

Initialization of the CRF transition matrix:

- The appropriate initialization of the transition matrix would be to choose small random values for all the cells of the transition matrix **except the cells representing transition from stop state, or transition to start state**. These cells should be initialized using a very high negative value (like -10000, -20000) so that the exponent of that tends to 0 basically suggesting that these transitions are impossible(logically we cannot transition to start state or transition from stop state, these have to be beginning and end states respectively).
- Another intuition is that if the forbidden transitions take place then, the large negative value of the transition score would make the exponent in the numerator of the likelihood function nearly 0, which would cause the negative log likelihood to go to +inf. To avoid this, these transitions would never take place.

Loss function for CRF:

- The negative of the log likelihood function described in the paper (Lample et.al.) was used as our loss function.

BiLSTM + CRF layer results

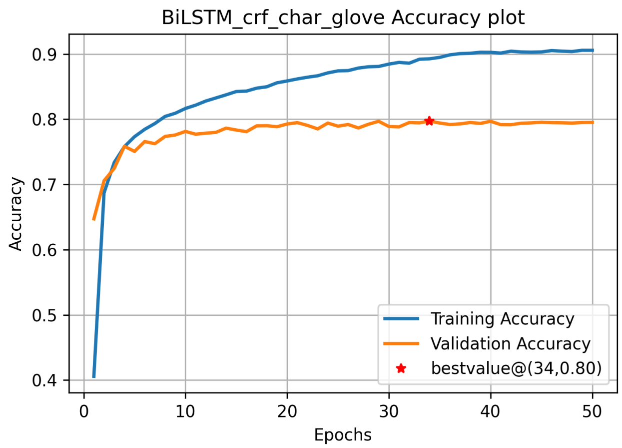
We experimented by first using out BiLSTM + CRF model with character level features and pretrained word embeddings which gave us a performance boost. After this we also tried out adding layer norm to the above architecture (using layer normalized BiLSTM) which further improved our performance. Results for both experiments are shown below

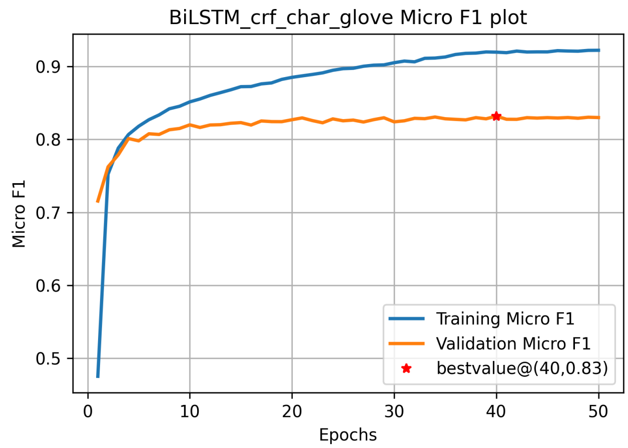
- **BiLSTM (Char level features + pretrained embeddings) + CRF**

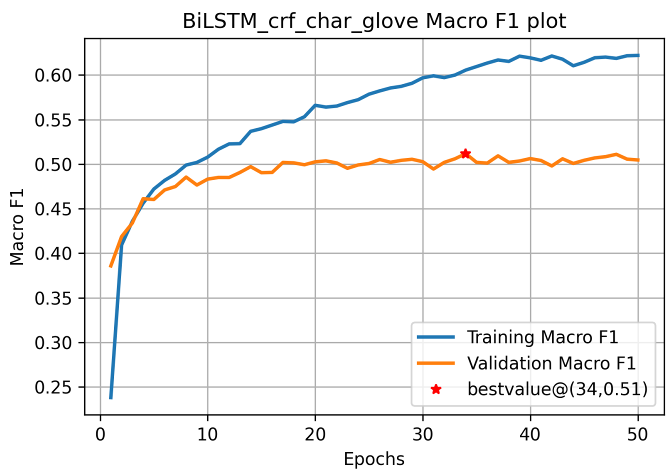
CONLL based final performance metrics (after training):

	Accuracy	**Micro F1**	**Macro F1**
Train	0.988	0.930	0.890
Validation	0.962	0.814	0.643
Test	0.962	0.817	0.634

Plots for performance statistics vs epochs:





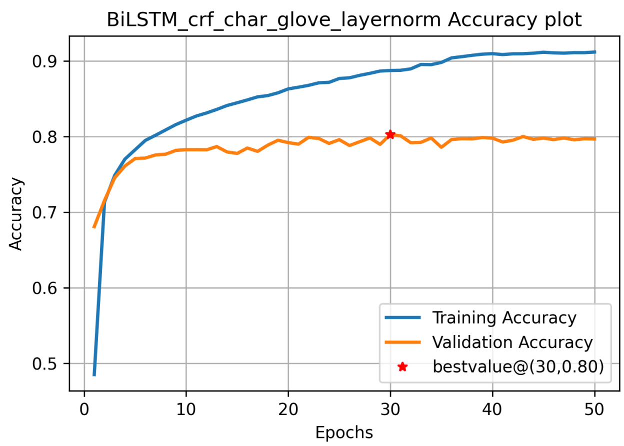


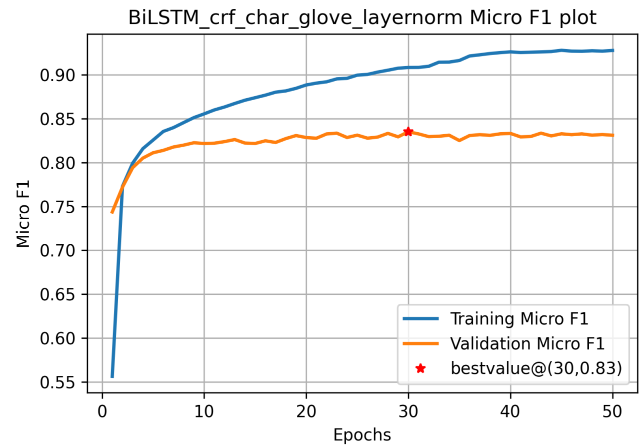
- **BiLSTM with Layer Normalization (Char level features + pretrained embeddings)+ CRF**

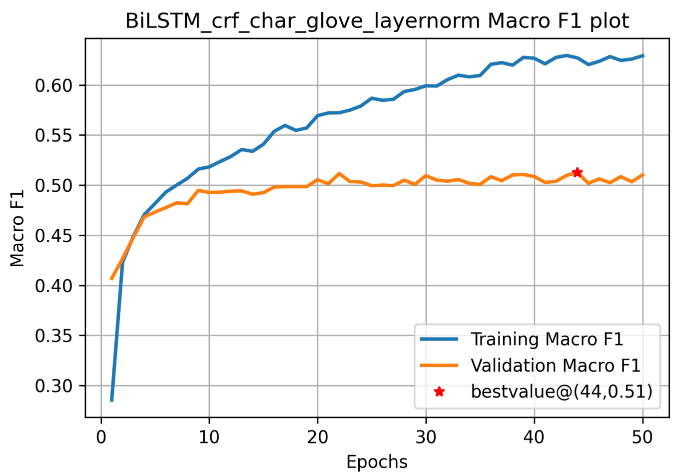
  CONLL based final performance metrics (after training):

	Accuracy	**Micro F1**	**Macro F1**
Train	0.990	0.941	0.907
Validation	0.964	0.820	0.649
Test	0.964	0.820	0.635

Plots for performance statistics vs epochs:







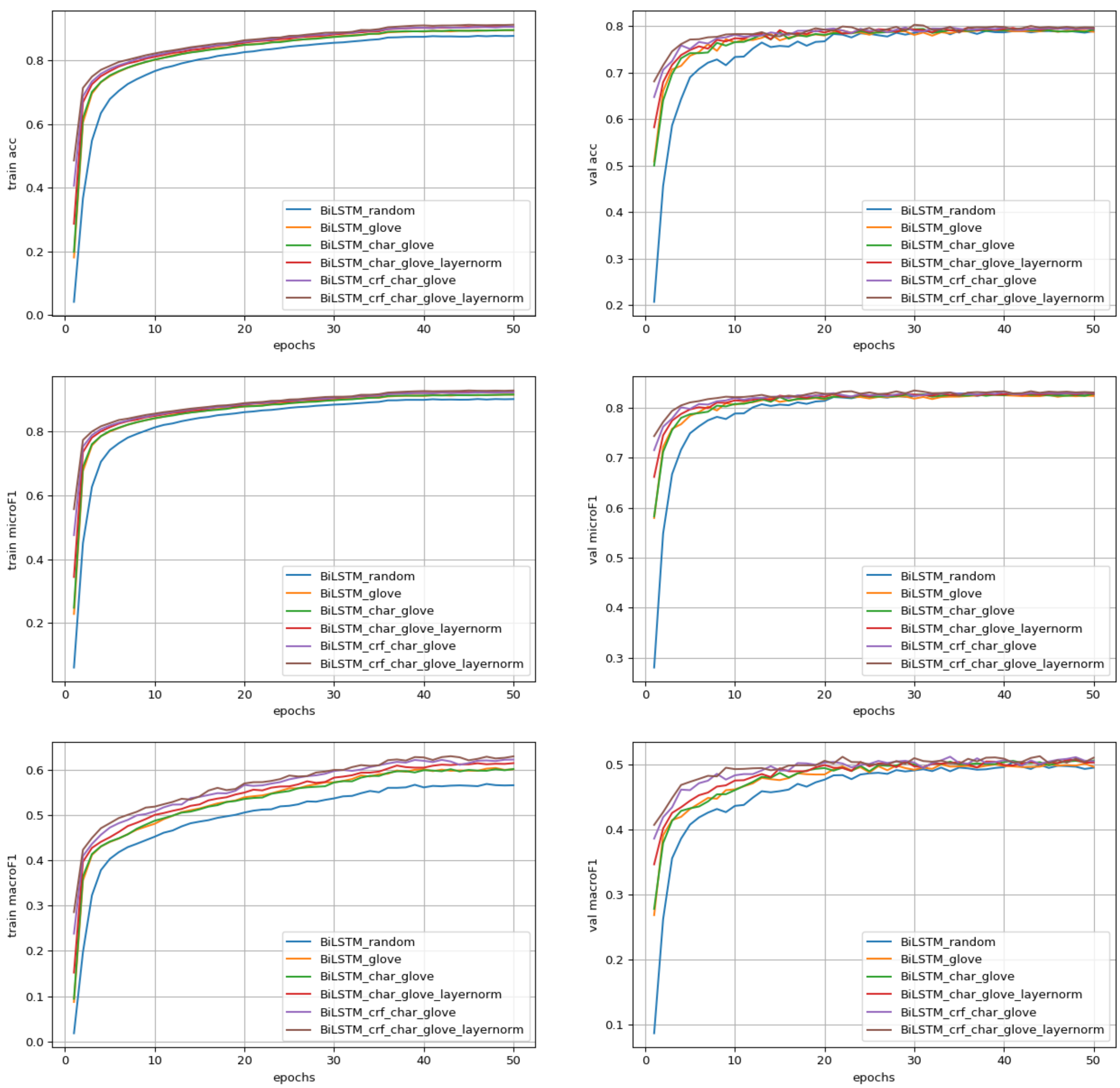
- **Comparison between BiLSTM with CRF with other variants**

On the basis of Final Performance:

Model	**Validation**		**Test**	
	MicroF1	**MacroF1**	**MicroF1**	**MacroF1**
BiLSTM (random embeddings)	0.803	0.619	0.804	0.614
BiLSTM (Glove embeddings)	0.793	0.620	0.794	0.614
BiLSTM (Glove + char embeddings)	0.795	0.629	0.795	0.626

BiLSTM with Layernorm (Glove + char embeddings)	0.799	0.625	0.802	0.627
BiLSTM with CRF (Glove + char embeddings)	**0.814**	**0.643**	**0.817**	**0.634**
BiLSTM with LayerNorm + CRF (Glove + char embeddings)	**0.820**	**0.649**	**0.820**	**0.635**

On the basis of learning curves:
(Following Plots, 3 rows respectively are for accuracy, microF1 and macroF1. Left column - Training, Right column - Validation)



Comments :

- The BiLSTM with CRF layer implemented on top of it, gives the best performance among all other variants for NER tagging. The gain in performance is also the highest, as compared to any other performance enhancer (layernorm/char embeddings/glove embeddings).

- This can be attributed to CRF being able to capture relationship between words using the transition scores internally, which can be thought of as a regularizer, which imposes (and learns) certain priors on the output sentences (whether one word can follow the other, etc)
- We see a gradual increase in performance as we increase the complexity of the model and introduce new performance enhancing measures. Glove embeddings proved to be better than random embeddings in terms of learning curves (faster learning). Including character level features improved performance and gave the model better representation of the tokens. This was improved upon by the normalization by the layernorm. Finally the CRF model with layer norm proved to be the best for ner tagging in our case.
- Referring to the curves above, we see that Bilstm with layernorm, crf and using char and glove embeddings learns the fastest among all models in general. The layernorm has its advantage in speed of training as described in the layernorm section of this document, and crf introduces extra constraints and learns the relationship between words in a sentence which may have helped us achieve faster training.
- Without layer norm, Bilstm with crf and using char and glove has slightly less speed of training and has the curve below that of the layernorm variant.
- The general trend of the plots is that the speed of training increases with increase in complexity of the model, and the complexity can be introduced by adding layernorm, crf, glove embeddings, char level embeddings etc. Best model is the combination of all, both in terms of speed and final accuracy.

Appendix

2.1.2

- Classification Report for Simple BiLSTM with Random Embeddings
  (Train, Validation and Test data respectively)

```
PERFORMANCE ON Train DATA
MicroF1 = 0.8891281071185972
Accuracy = 0.9821888393554009
------------Classification Report-------------
              precision    recall  f1-score   support

         art      0.673     0.341     0.453       296
         eve      0.543     0.478     0.508       226
         geo      0.886     0.937     0.911     29240
         gpe      0.968     0.942     0.955     12058
         nat      0.714     0.564     0.630       133
         org      0.805     0.777     0.791     15803
         per      0.879     0.879     0.879     13121
         tim      0.922     0.911     0.917     15767

   micro avg      0.887     0.891     0.889     86644
   macro avg      0.799     0.729     0.755     86644
weighted avg      0.886     0.891     0.888     86644
```

```
PERFORMANCE ON Validation DATA
MicroF1 = 0.8034243979743
Accuracy = 0.9635724503487508
------------Classification Report-------------
              precision    recall  f1-score   support

         art      0.265     0.086     0.129       105
         eve      0.380     0.244     0.297        78
         geo      0.823     0.866     0.844      9724
         gpe      0.936     0.908     0.922      4210
         nat      0.629     0.440     0.518        50
         org      0.635     0.623     0.629      5187
         per      0.757     0.744     0.750      4457
         tim      0.881     0.853     0.867      5254

   micro avg      0.804     0.802     0.803     29065
   macro avg      0.663     0.595     0.619     29065
weighted avg      0.803     0.802     0.802     29065
```

```
PERFORMANCE ON Test DATA
MicroF1 = 0.8038678760910491
Accuracy = 0.963636161631873
------------Classification Report-------------
              precision    recall  f1-score   support

         art      0.280     0.069     0.110       102
         eve      0.378     0.322     0.348        87
         geo      0.831     0.871     0.851      9912
         gpe      0.928     0.907     0.918      4168
         nat      0.571     0.364     0.444        55
         org      0.634     0.623     0.628      5205
         per      0.758     0.753     0.756      4406
         tim      0.868     0.844     0.856      5275

   micro avg      0.804     0.804     0.804     29210
   macro avg      0.656     0.594     0.614     29210
weighted avg      0.802     0.804     0.802     29210
```

- Classification Report for Simple BiLSTM with Glove

(Train, Validation and Test data respectively)

```
PERFORMANCE ON Train DATA
MicroF1 = 0.9092593551654092
Accuracy = 0.9859880455904748
------------Classification Report-------------
              precision    recall  f1-score   support

         art      0.782     0.632     0.699       296
         eve      0.716     0.681     0.698       226
         geo      0.901     0.949     0.925     29240
         gpe      0.970     0.952     0.961     12058
         nat      0.728     0.564     0.636       133
         org      0.848     0.809     0.828     15803
         per      0.908     0.902     0.905     13121
         tim      0.930     0.935     0.933     15767

   micro avg      0.907     0.912     0.909     86644
   macro avg      0.848     0.803     0.823     86644
weighted avg      0.906     0.912     0.909     86644
```

```
PERFORMANCE ON Validation DATA
MicroF1 = 0.7928936234766631
Accuracy = 0.9614248697959983
------------Classification Report-------------
              precision    recall  f1-score   support

         art      0.500     0.152     0.234       105
         eve      0.400     0.282     0.331        78
         geo      0.807     0.872     0.838      9724
         gpe      0.934     0.914     0.924      4210
         nat      0.514     0.380     0.437        50
         org      0.583     0.638     0.609      5187
         per      0.737     0.735     0.736      4457
         tim      0.844     0.866     0.855      5254

   micro avg      0.777     0.809     0.793     29065
   macro avg      0.665     0.605     0.620     29065
weighted avg      0.779     0.809     0.793     29065
```

```
PERFORMANCE ON Test DATA
MicroF1 = 0.7943733843623058
Accuracy = 0.9621140577955047
------------Classification Report-------------
              precision    recall  f1-score   support

         art      0.480     0.118     0.189       102
         eve      0.341     0.322     0.331        87
         geo      0.818     0.875     0.846      9912
         gpe      0.931     0.916     0.923      4168
         nat      0.621     0.327     0.429        55
         org      0.585     0.634     0.609      5205
         per      0.735     0.743     0.739      4406
         tim      0.836     0.860     0.848      5275

   micro avg      0.779     0.810     0.794     29210
   macro avg      0.668     0.599     0.614     29210
weighted avg      0.780     0.810     0.794     29210
```

2.1.3

- Classification Report for Simple BiLSTM with Character + Word embeddings

(Train, Validation and Test data respectively)

```
PERFORMANCE ON Train DATA
MicroF1 = 0.9083878233489238
Accuracy = 0.9860040241349769
------------Classification Report-------------
              precision    recall  f1-score   support

         art      0.738     0.581     0.650       296
         eve      0.738     0.673     0.704       226
         geo      0.899     0.952     0.925     29240
         gpe      0.972     0.949     0.960     12058
         nat      0.741     0.602     0.664       133
         org      0.850     0.816     0.832     15803
         per      0.905     0.896     0.900     13121
         tim      0.923     0.934     0.928     15767

   micro avg      0.904     0.912     0.908     86644
   macro avg      0.846     0.800     0.820     86644
weighted avg      0.904     0.912     0.908     86644
```

```
PERFORMANCE ON Validation DATA
MicroF1 = 0.7951207958313595
Accuracy = 0.9612918011214732
------------Classification Report-------------
              precision    recall  f1-score   support

         art      0.370     0.162     0.225       105
         eve      0.469     0.295     0.362        78
         geo      0.809     0.875     0.841      9724
         gpe      0.938     0.910     0.924      4210
         nat      0.512     0.420     0.462        50
         org      0.563     0.640     0.599      5187
         per      0.774     0.729     0.751      4457
         tim      0.870     0.862     0.866      5254

   micro avg      0.782     0.808     0.795     29065
   macro avg      0.663     0.612     0.629     29065
weighted avg      0.786     0.808     0.796     29065
```

```
PERFORMANCE ON Test DATA
MicroF1 = 0.7945924132364811
Accuracy = 0.9616400205909911
------------Classification Report-------------
              precision    recall  f1-score   support

         art      0.308     0.118     0.170       102
         eve      0.444     0.368     0.403        87
         geo      0.818     0.880     0.848      9912
         gpe      0.931     0.911     0.921      4168
         nat      0.600     0.382     0.467        55
         org      0.564     0.636     0.598      5205
         per      0.766     0.734     0.749      4406
         tim      0.852     0.853     0.852      5275

   micro avg      0.781     0.809     0.795     29210
   macro avg      0.660     0.610     0.626     29210
weighted avg      0.784     0.809     0.795     29210
```

2.1.4

- Classification Report for BiLSTM with Layer Normalization using Character + Word embeddings (Train, Validation and Test data respectively)

```
PERFORMANCE ON Train DATA
MicroF1 = 0.9233053568152632
Accuracy = 0.9887547920269522
------------Classification Report-------------
              precision    recall  f1-score   support

         art      0.779     0.713     0.744       296
         eve      0.683     0.752     0.716       226
         geo      0.915     0.958     0.936     29240
         gpe      0.972     0.955     0.964     12058
         nat      0.673     0.789     0.727       133
         org      0.875     0.848     0.861     15803
         per      0.918     0.917     0.917     13121
         tim      0.931     0.955     0.943     15767

   micro avg      0.917     0.929     0.923     86644
   macro avg      0.843     0.861     0.851     86644
weighted avg      0.917     0.929     0.923     86644

PERFORMANCE ON Validation DATA
MicroF1 = 0.7989299742652038
Accuracy = 0.9626003097543034
------------Classification Report-------------
              precision    recall  f1-score   support

         art      0.385     0.190     0.255       105
         eve      0.289     0.282     0.286        78
         geo      0.810     0.874     0.841      9724
         gpe      0.940     0.912     0.926      4210
         nat      0.460     0.460     0.460        50
         org      0.597     0.643     0.620      5187
         per      0.767     0.736     0.752      4457
         tim      0.850     0.870     0.860      5254

   micro avg      0.786     0.812     0.799     29065
   macro avg      0.637     0.621     0.625     29065
weighted avg      0.788     0.812     0.799     29065
```

```
PERFORMANCE ON Test DATA
MicroF1 = 0.801542399137873
Accuracy = 0.9631102766081157
------------Classification Report-------------
              precision    recall  f1-score   support

         art      0.245     0.118     0.159       102
         eve      0.361     0.402     0.380        87
         geo      0.822     0.881     0.850      9912
         gpe      0.936     0.910     0.923      4168
         nat      0.535     0.418     0.469        55
         org      0.598     0.645     0.621      5205
         per      0.764     0.746     0.755      4406
         tim      0.845     0.866     0.855      5275

   micro avg      0.789     0.815     0.802     29210
   macro avg      0.638     0.623     0.627     29210
weighted avg      0.790     0.815     0.802     29210
```

2.2

- Classification Report for BiLSTM + CRF with Character + Word embeddings
  (Train, Validation and Test data respectively)

```
PERFORMANCE ON Train DATA
MicroF1 = 0.93000374866634
Accuracy = 0.9880861513954801
------------Classification Report-------------
              precision    recall  f1-score   support

         art      0.912     0.767     0.833       296
         eve      0.846     0.827     0.837       226
         geo      0.923     0.958     0.940     29240
         gpe      0.975     0.961     0.968     12058
         nat      0.775     0.827     0.800       133
         org      0.887     0.850     0.868     15803
         per      0.934     0.920     0.927     13121
         tim      0.947     0.951     0.949     15767

   micro avg      0.929     0.931     0.930     86644
   macro avg      0.900     0.883     0.890     86644
weighted avg      0.929     0.931     0.930     86644
```

```
PERFORMANCE ON Validation DATA
MicroF1 = 0.8143543471148381
Accuracy = 0.9615875092870846
------------Classification Report-------------
              precision    recall  f1-score   support

         art      0.500     0.171     0.255       105
         eve      0.423     0.282     0.338        78
         geo      0.828     0.870     0.848      9724
         gpe      0.939     0.921     0.930      4210
         nat      0.524     0.440     0.478        50
         org      0.645     0.656     0.650      5187
         per      0.803     0.740     0.770      4457
         tim      0.884     0.866     0.875      5254

   micro avg      0.815     0.814     0.814     29065
   macro avg      0.693     0.618     0.643     29065
weighted avg      0.815     0.814     0.813     29065
```

```
PERFORMANCE ON Test DATA
MicroF1 = 0.816718054128126
Accuracy = 0.9620807270545624
------------Classification Report-------------
              precision    recall  f1-score   support

         art      0.414     0.118     0.183       102
         eve      0.417     0.345     0.377        87
         geo      0.838     0.877     0.857      9912
         gpe      0.931     0.917     0.924      4168
         nat      0.478     0.400     0.436        55
         org      0.651     0.656     0.654      5205
         per      0.801     0.748     0.774      4406
         tim      0.879     0.862     0.870      5275

   micro avg      0.817     0.816     0.817     29210
   macro avg      0.676     0.615     0.634     29210
weighted avg      0.816     0.816     0.816     29210
```

- Classification Report for BiLSTM + CRF with Character + Word embeddings AND Layer Norm (Train, Validation and Test data respectively)

PERFORMANCE ON Train DATA
MicroF1 = 0.9408832687736344
Accuracy = 0.9901621330619439
------------Classification Report-------------

	precision	recall	f1-score	support
art	0.900	0.818	0.857	296
eve	0.858	0.832	0.845	226
geo	0.933	0.968	0.950	29240
gpe	0.977	0.967	0.972	12058
nat	0.800	0.902	0.848	133
org	0.919	0.869	0.893	15803
per	0.934	0.937	0.936	13121
tim	0.944	0.964	0.954	15767
micro avg	0.938	0.943	0.941	86644
macro avg	0.908	0.907	0.907	86644
weighted avg	0.938	0.943	0.941	86644

PERFORMANCE ON Validation DATA
MicroF1 = 0.8200622621930128
Accuracy = 0.964174955736184
------------Classification Report-------------

	precision	recall	f1-score	support
art	0.346	0.171	0.229	105
eve	0.450	0.346	0.391	78
geo	0.834	0.868	0.851	9724
gpe	0.931	0.923	0.927	4210
nat	0.480	0.480	0.480	50
org	0.714	0.648	0.679	5187
per	0.784	0.754	0.769	4457
tim	0.865	0.873	0.869	5254
micro avg	0.824	0.816	0.820	29065
macro avg	0.676	0.633	0.649	29065
weighted avg	0.821	0.816	0.818	29065

PERFORMANCE ON Test DATA
MicroF1 = 0.8200697390796503
Accuracy = 0.9640398339388417
-----------Classification Report-------------

	precision	recall	f1-score	support
art	0.233	0.098	0.138	102
eve	0.397	0.356	0.376	87
geo	0.840	0.871	0.855	9912
gpe	0.924	0.924	0.924	4168
nat	0.464	0.473	0.468	55
org	0.705	0.644	0.673	5205
per	0.784	0.767	0.775	4406
tim	0.863	0.871	0.867	5275
micro avg	0.823	0.817	0.820	29210
macro avg	0.651	0.625	0.635	29210
weighted avg	0.819	0.817	0.818	29210