Project - 2: Neural Conditional Random Fields for Named Entity Recognition

Anonymous ACL-IJCNLP submission

Abstract

In this report, I have implemented BiLST-CRF along with veterbi algorithm for decoding. As a part of Optional task, I have experimented with BERT word embeddings. I have also added Capitalization as a feature in the BiLSTM-CRF model which has led to significant improvement in the F1, precision and recall metrics. Apart from this, I have conducted analysis on out of vocabulary words. All of the span level, token level and Out of Vocabulary words analysis are included in this report.

1 Implementation choices

In the assignment, I have built on the starter code provided. I have added functions in the class to compute the partition function and also modified the loss function to implement the negative log likelihood loss. I have computed the **log_sum_exp** for each input in one shot, rather than iterating over all possible output tokens. Viterbi algorithm also has been implemented for decoding.

I have implemented a BiLSTM-CRF model with a single layer. The architecture is same as described in the assignment's problem statement. The work embedding dimension is 128 and dimension of each hidden layer is 256. A dropuout of 0.3 is used for regularization and adam optimizer has been used with a learning rate of 0.003. The model converges within 10 epochs and it takes around 15 minutes to complete the training.

2 Performance

2.1 Span level performance

In figure 1, where we plot the loss curve of BiLSTM-CRF model. Here we can observe that the validation loss increases after 10 epochs, this signifies that the model is overfitting. We use early stopping in the analysis as an regularization method to avoid overfitting. All the analysis in the report are computed after training the BiLSTM-CRF model upto 10 epochs only.

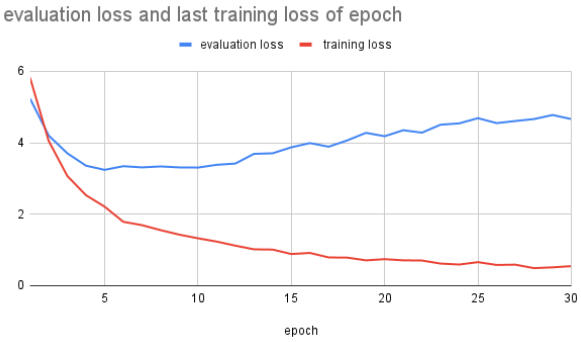

Figure 1: Loss Curve of BiLSTM-CRF

In figure 2, we plot loss curve of vanilla BiLSTM model upto 10 epochs. We can observe that the training loss and validation loss have reached a plateau.

Figure 2: Loss curve BiLSTM

In table 1, we can see that the BiLSTM model marginally outperforms in terms of accuracy in identifying non-O tags. Although the precision and FB1 values of both the models across all tags is almost the same, difference arise when we look at tag level metrics. The BiLSTM considerably performs better in terms of recall for all the tags.

This signals that the number of false negatives have reduced significantly. The BiLSTM-CRF model uses a transition matrics in CRF computation. Intuitively, this metrics learns the grammar and structure of the input sentences. This helps the CRF to correctly classify the false negatives from BiLSTM models. This can be illustrated with the help of these examples:

Example 1: *A hijacked Sudan Airways plane with 100 passengers and crew on board was expected to land at London 's Stansted airport later on Tuesday morning , a police spokeswoman said .*
Highlighted Spans: *Sudan Airways* and *London 's Stansted airport*
BiLSTM's output *'I-LOC', 'I-LOC'* and *'I-LOC', 'I-LOC', 'I-LOC'*
BiLSTM-CRF's output *'I-ORG', 'I-ORG'* and *'I-LOC', 'O', 'I-LOC'*
Explanation In this example the CRF correctly classifies Sudan Airways as Organization and London Stansted airport as location. The CRF is also correctly able to identify that $'s$ is not a location.

Example 2 *Nice are 8th in the table*
Highlighted Spans: *Nice*
BiLSTM's output *'I-LOC'*
BiLSTM-CRF's output *'I-ORG'*
Explanation Although Nice is a place in France, in this context it refers to a football club, hence an organization. The CRF is able to correctly guess this information from the context.

2.2 Token Level performance

If we compare span level and token level performance on all the tags (table 2 and table 1), we can observe that all the metrics (precision, recall, FB1) are considerably higher for the later. Also, BiLSTM-CRF outperforms BiLSTM on all the fronts. We can conclude that if single tokens are considered BiLSTM-CRF correctly identifies them, but when compared with the spans they miss some tags. This can be due to two reasons, firstly not enough context present in the training data and secondly, poor generalization of the trained model.

Example 3 *The News Agency of Nigeria, NAN*
Expected output *'O', 'I-ORG', 'I-ORG', 'I-ORG', 'I-ORG'*
BiLSTM-CRF's output *'O', 'O', 'O', 'O', 'I-LOC', 'O', 'I-ORG'*
BiLSTM's output *'O', 'O', 'O', 'O', 'I-LOC', 'O', 'O'*
Explanation The training data does not contain the instance of News Agency of Nigeria, the model has only seen instances of Nigeria as location, but correctly identifies *NAN* as organization even though it was out of vocabulary. The model is able to identify the grammar, but is not able to generalize well on unseen instances. If we contrast it with BiLSTM's output, it incorrectly states *NAN* as O. This demonstrates that the CRF addition to the BiLSTM has helped it to understand the structure of the sentence better.

Due to these kind of errors mentioned in the above paragraphs, the span level metrics are lower as compared to the tag level metrics. Partial identification of spans penalizes the precision and recall, increases these metrics when measured individually.

2.3 Out of vocabulary words

The goal of the language models is to learn structure of the sentence is such a manner that it performs well in cases which it has not encountered before. In the subsection, I analyze the model's performance on OOV words. From, figure 3 and 4 we can conclude that the BiLSTM-CRF is able to classify more OOV words correctly. If we compare recall for any token wrt total (for eg for LOC recall is 37.62 vs 79.61), we observe a huge difference. The model's heavily underperforms in case of out of vocabulary words and if we can improve our strategy, then a better performance can be observed here.

BiLSTM-CRF	precision	recall	f1
O	81.71	90.22	85.75
I-PER	76.99	79.73	78.34
I-ORG	62.62	53.09	57.46
I-LOC	73.08	37.62	49.67
I-MISC	38.89	7.61	12.73

Figure 3: OOV performance of BiLSTM CRF

3 Error Analysis and Observations

As shown in the previous section the BiLSTM-CRF is able to learn additional structural properties from the training dataset. It also heavily underperforms in case of out of vocabulary words. Also,

Tag Type	Model Type	Precision	Recall	FB1 score
PER	BiLSTM	69.79	72.63	71.18
	BiLSTM-CRF	68.27	72.9	70.51
	BERT-BiLSTM-CRF	71.39	68.29	69.81
ORG	BiLSTM	67.16	58.63	62.61
	BiLSTM-CRF	61.84	57	59.32
	BERT-BiLSTM-CRF	52.6	32.9	40.48
MISC	BiLSTM	78.72	57.81	66.67
	BiLSTM-CRF	75.5	59.38	66.47
	BERT-BiLSTM-CRF	69.9	37.5	48.81
LOC	BiLSTM	86.48	75.76	80.76
	BiLSTM-CRF	88.65	79.61	83.89
	BERT-BiLSTM-CRF	80.56	63.91	71.27
Overall	BiLSTM	75.07	67.75	71.22
	BiLSTM-CRF	73.4	68.81	71.03
	BERT-BiLSTM-CRF	70.19	53.37	60.64
Accuracy				
Overall Non O Tag	BiLSTM	72.03		
	BiLSTM-CRF	73.79		
	BERT-BiLSTM-CRF	61.84		
Overall Combined Tag	BiLSTM	94.23		
	BiLSTM-CRF	94.37		
	BERT-BiLSTM-CRF	92.79		

Table 1: **Span level analysis**

Tag Type	Model Type	Precision	Recall	FB1 score
PER	BiLSTM	82.07	83.23	82.65
	BiLSTM-CRF	83	84.17	83.58
	BERT-BiLSTM-CRF	86	80.88	83.36
ORG	BiLSTM	81.02	61.84	70.14
	BiLSTM-CRF	76.2	64.69	69.98
	BERT-BiLSTM-CRF	75.25	45.92	57.03
MISC	BiLSTM	84.75	57.03	68.18
	BiLSTM-CRF	83.15	56.27	67.12
	BERT-BiLSTM-CRF	83.19	37.64	51.83
LOC	BiLSTM	88.28	77.14	82.34
	BiLSTM-CRF	89.89	80.48	84.92
	BERT-BiLSTM-CRF	84.57	65.24	73.66
Overall	BiLSTM	95.96	98.54	97.23
	BiLSTM-CRF	96.34	98.36	97.34
	BERT-BiLSTM-CRF	94.13	98.9	96.46

Table 2: **Token level analysis**

BiLSTM	precision	recall	f1
O	81.05	90.85	85.67
I-PER	76.38	79.96	78.13
I-ORG	69.83	51.44	59.24
I-LOC	75.51	36.63	49.33
I-MISC	52.63	10.87	18.02

Figure 4: OOV performance of BiLSTM

the BiLSTM-CRF model is efficiently able to identify individual tokens, but in few cases fails to identify the complete phrase successfully.

Example 4 *At Portsmouth :* $< unk >$ *0000 in 0000 overs*
Unknown token: *Middlesex*
Expected output *'O', 'I-LOC', 'O', 'I-ORG', 'O', 'O', 'O', 'O'*
BiLSTM-CRF's output *'O', 'I-LOC', 'O', 'I-ORG', 'O', 'O', 'O', 'O'*
BiLSTM's output *'O', 'I-LOC', 'O', 'I-LOC', 'O', 'O', 'O', 'O',*
Explanation Middlesex is a cricket team, therefore an organization. The BiLSTM-CRF model identifies that it is a game and correctly tags it as an organization, whereas BiLSTM model tags it as location.

The poor performance of all the models on MISC tag can be attributed to very less number of training examples in the dataset. If we introduce new examples in the dataset then the performance would improve on this tag.

4 Pre-trained language model

I have used word embeddings of pretrained *bert-base-uncased* model. To obtain the embeddings, I have summed up the weights of the last 4 hidden layers of the model. The length of the embedding is 768. In this experiment the idea is to use a pretrained model's embeddings instead of learning them from scratch. I have used a similar setup for this experiment as in BiLSTM-CRF case. The training time had increased, and due to that I was not able to completely train the model. Even though the model I observe that the model heavily outperforms in every metrics. I have done a comparative study between this case and the results can be found in table 1 and 2. I have compared the tag level and overall metrics in span level and tag level case. We can observe that this model beats the other mod in every scenario. This can be attributed to the fact that BERT model has been trained on huge corpus of data and has learned deeper and more nuanced context when compared to learning the word embeddings by scratch.

5 Capitalization as a feature

I incorporated capitalization as a binary feature in the model. If the word in the input sentence in capital, I have passed it as 1 otherwise 0. This has lead to significant improvement in the metrics. Since this is a low resource training environment, adding handcrafted features had a huge effect on the performance of the model and also generalization on the test dataset. In such cases, the model cannot be made complex enough to automatically learn minute nuances, therefore adding the quality features such as capitalization, word-form or presence of a word in a gazetteer have positive affect on the model performance. The performance on the validation set can be observed in figure 5 and the performance has been mentioned in firgure 4.

```
processed 11170 tokens with 1231 phrases; found: 1297 phrases; correct: 925.
accuracy:  78.47%; (non-O)
accuracy:  95.55%; precision:  71.32%; recall:  75.14%; FB1:  73.18
             LOC: precision:  84.46%; recall:  82.37%; FB1:  83.40  354
            MISC: precision:  78.53%; recall:  66.67%; FB1:  72.11  163
             ORG: precision:  54.86%; recall:  68.08%; FB1:  60.76  381
             PER: precision:  72.43%; recall:  78.32%; FB1:  75.26  399
```

Figure 5: Capitalization as a feature

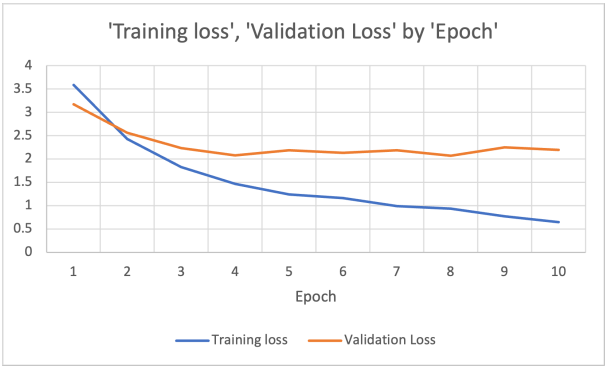

Figure 6: Training & Validation loss for Capitalization

Capitalization has helped the model to correctly predict many instances of location, person and organization which were earlier classified as *O*. Capitalization is also able to correct many errors in the valnilla CRF models pertaining to misclassification of "O". For example,

Example 5 *Peru 's guerrillas* $< unk >$ *one , take 0000* $< unk >$ *in* $< unk >$.
Error token "in" is tagged as 'I-LOC' which is incorrect, capitalization correctly tags it as *O*

6 Conclusion

In this report I have analysed, described and reported the efficacy of BiLSTM-CRF techniques over prediction of location, organization, person and miscellaneous tags. It is also seen that adding features such as capitalization can drastically improve the performance of the model.

7 References

[1] Zhiheng Huang, Wei Xu, and Kai Yu. Bidirectional lstm-crf models for sequence tagging. arXiv preprint arXiv:1508.01991. 2015.
[2] John Lafferty, Andrew McCallum, Fernando Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. Proc. 18th International Conf. on Machine Learning. Morgan Kaufmann. pp. 282–289. 2001.
[3] http://pytorch.org/tutorials/beginner/nlp/advanced_tutorial.html