# Neural CRF for Named Entity Recognition

**Murali Mohana Krishna Dandu**
UC San Diego (PID: A59004607)
mdandu@ucsd.edu

## Abstract

This documents details the implementation results of a Bi-LSTM CRF model for Named Entity Recognition using the CoNLL dataset. The Forward algorithm for global normalization and the Viterbi algorithm for decoding were implemented. The model was compared against the Bi-LSTM baseline and has been evaluated both qualitatively and quantitatively.

## 1 Introduction

Named Entity Recognition (NER) is modelled as a tagging problem using BIO tags across the tokens of the sentence. RNNs have been very powerful in Seq-to-Seq labelling and have been used for tagging problems like NER and POS. However, one drawback is that the independent classification layers on top of RNN features are locally optimized i.e., they don't learn the dependency structure between the label tags. If we try to address this by passing the previous timestep output to the current timestep, it suffers as the prediction errors can propagate throughout the sequence. To remedy this, we use a Conditional Random Field (CRF) layer on top of RNN to incorporate the dependency structure between neighboring output labels.

## 2 Approach

This section briefly discusses the network architecture of the Bi-LSTM CRF and the two core components - Forward and Viterbi algorithms.

### 2.1 Conditional Random Fields

CRF is a globally normalized discriminative model that estimates the probability of the entire sequence p(Y/X) in a log-linear way.

$$p(Y|X;\theta) = \frac{\Phi(X,Y)}{\sum_{Y' \in \bigcirc} \Phi(X,Y')}$$

In particular, the linear-chain CRF feature functions depends on the whole input, the current timestep, the current label and the previous label. Traditionally these functions are handcrafted based on multiple linguistic patterns, however, we learn these features in a neural network framework. The feature functions are a product of emission and transition potentials. The BiLSTM output features are used as log emission potentials and the transition weight matrix over the entire tag set is used as log of transition potentials.

$$\Phi(X,Y) = \prod_{t=1}^{T} \phi_{em}(X, y_t) * \phi_{tr}(y_{t-1}, y_t)$$

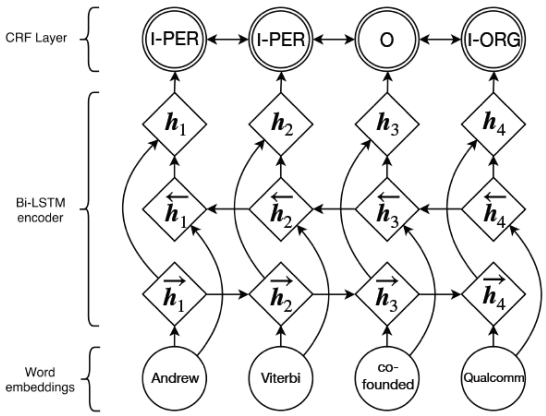


Figure 1: BiLSTM CRF Architecture

### 2.2 Forward Algorithm

The denominator in the probability is also called partition function which iterates over all the possible sequence tags and is very expensive to calculate. Hence, we use a dynamic programming approach which stores the sum of forward scores at each timestep and recurs through the entire sequence.

**Algorithm 1** CRF Forward Algorithm

```
Initialize start_tag forward scores = 0
for every feature across sentence do
    for every tag across tagset do
        tag score = forward + trans + emit
        tag score = logsumexp(tag score)
    end for
    Append tag score to forward matrix
end for
Add stop_tag transition to forward score
Return logsumexp(forward score)
```

### 2.3 Viterbi Decoding Algorithm

Once the model is built and the emission and transition features are learnt, we need to identify the best tagset combination that maximizes the global probability for a given input sequence. Viterbi algorithm is used here to calculate the argmax of best path for every timestep-tagset combination matrix. Back-pointer matrix is used to store which tag from previous timestep is giving the maximum forward score and traced back to identify the best tagset sequence.

**Algorithm 2** CRF Viterbi Decoding Algorithm

```
Initialize start_tag viterbi scores = 0
Initialize backpointer matrix
for every feature across sentence do
    for every tag across tagset do
        best tag = argmax(forward + trans)
    end for
    Append best tag to backpointer
end for
Append stop_tag score argmax to bkptr
Initialize best path
for reverse backpointer do
    Append best path id from bkptr matrix
end for
Return the reverse best path
```

## 3 Results and Discussion

### 3.1 Overview

The above BiLSTM and BiLSTM-CRF is trained for 30 epochs using the following hyperparameters: 128 embedding dim, 256 hidden dim and dropout probability of 0.3. Fig. 2 shows the training and evaluation loss for both the models. We can see that the training loss gradually decreases and stabilizes after 20 epochs. However, the evaluation loss starts overfitting after 10 epochs due to lesser training data size. The important point to note here is that the CRF model evaluation loss is consistently lower compared to the baseline suggesting that the global normalization using the linear chain CRF improved the results. This trend is also seen in Fig. 3 where the Precision, Recall and F1 scores are almost consistently better after 5 epochs.

I took the models with best evaluation F1 score (in both the baseline and the CRF) for further quantitative and qualitative analysis.

| | **LSTM** | **LSTM+CRF** |
|---|---|---|
| Non-O Accuracy | 75.83% | **77.09%** |
| Accuracy | 94.55% | 94.74% |
| Span F1 | 72.02 | **74.59** |

Table 1: Evaluation Metrics

| | **Precision** | | **Recall** | | **F1** | |
|---|---|---|---|---|---|---|
| | **L** | **L+CRF** | **L** | **L+CRF** | **L** | **L+CRF** |
| Total | 73.8% | **77.0%** | 70.4% | 72.4% | 72.0 | **74.6** |
| LOC | 86.5% | 88.5% | 81.3% | 82.4% | 83.8 | **85.3** |
| MISC | 79.4% | 77.5% | 56.3% | 57.3% | 65.9 | 65.9 |
| ORG | 69.3% | 70.7% | 59.6% | **64.5%** | 64.1 | **67.5** |
| PER | 64.7% | **71.4%** | 75.9% | 77.0% | 69.9 | **74.0** |

Table 2: Tag-wise Span-level Evaluation Metrics

We can see from Table 1 that there is 1.5% improvement in Non-O accuracy and 2.5% improvement in span level F1 score. Table 2 shows that we achieved 2-5% F1 improvement across LOC, ORG and PER tags and the scores remained same for the MISC class. We can also see that the greater value in CRF model is it's improved Precision compared to Recall. Analyzing results across the tags, we see that MISC and ORG tags perform low compared to other tags. I performed token level analysis and OOV analysis to understand the results further.

### 3.2 Token Level Analysis

| | | **Token F1** | |
|---|---|---|---|
| | **Tokens** | **L** | **L+CRF** |
| Total | 1816 | 86.0 | **86.6** |
| O | 424 | 97.3 | 97.5 |
| LOC | 424 | 83.2 | **86.0** |
| MISC | 264 | 66.2 | 66.2 |
| ORG | 490 | 73.4 | 70.1 |
| PER | 638 | 81.8 | **84.6** |

Table 3: Tag-wise Token-level Evaluation Metrics

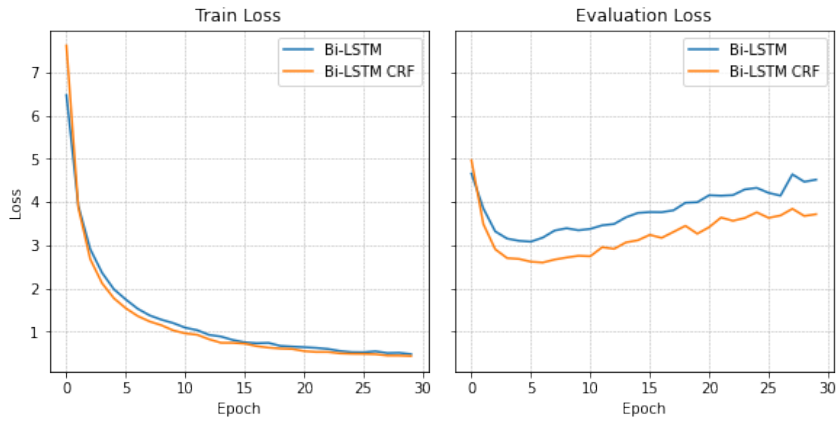


Figure 2: Training and Evaluation Loss

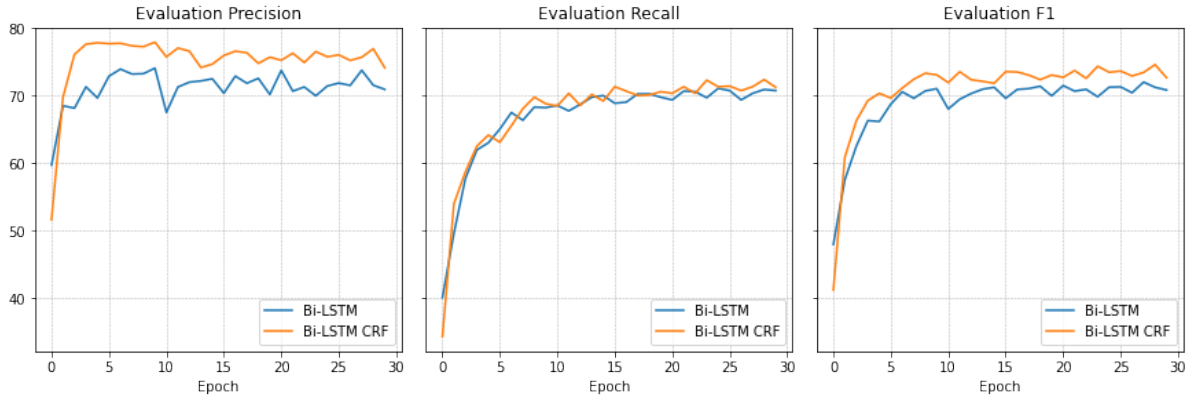


Figure 3: Evaluation Metrics

Table 3 shows the F1 scores calculated at token level without taking the span/chunk correctness into account. (Please note that the 'Total F1' is calculated using only O/Non-O tags in Table 3). It follows a similar trend as span-F1 but the increment compared to baseline model is lower. Similar to span-level metrics, the LOC and PER tags sees 3% improvement but the overall improvement is under 1%. One notable difference is that we see a 3% reduction is ORG token-level F1 compared to 3% improvement in span-level F1, suggested that ORG spanning across multiple tokens are predicted much better in CRF model compared to single token ORG. Overall, the CRF model is bringing value in predicting the entire span/chunk correctly, which is the main objective of structured prediction.

### 3.3 Samples Comparison

Table 4 shows some sample predictions from both the models. We can see that the CRF model accurately predicts spans where there is <unk> token present like *'T.'*, *'Spain'* etc., compared to the baseline model. The tag dependency structure in the CRF model can be helping predict these unknowns better. You can also see that *"s'* is accurately predicted maybe using the preceding *'Lloyd'* label 'I-ORG'. However, in some cases, the relationship is learnt in an ambiguous way and may be acting negatively. For example, in the *'New York Newsdesk'* and *'Ford China JV'*, even though it is semi-accurate, the last token in the span is incorrectly predicted. Connecting to the above class level metrics, we see that most of the ambiguities occur in ORG/MISC classes and the main reason for their lower F1 scores.

### 3.4 OOV Analysis

Predicting unknown vocabulary accurately is one of the key aspects in NER. As the written and spoken vocabulary is growing at fast pace, as well as the dominant paradigm of transfer learning in NLP make the OOV analysis crucial in evaluating NER systems. To understand this, I conducted couple of analysis. Fist, I separated the results of sentences where there is at least one <unk> token present versus where there is none. This gives us an idea on

| Type | Sentence/Prediction |
|---|---|
| SRC | [**T.**, Moody, not, out, 0000] |
| SRC Token | [<**unk**>, Moody, not, out, 0000] |
| Truth | [I-PER, I-PER, O, O, O] |
| LSTM | [O, I-PER, O, O, O] |
| LSTM+CRF | [I-PER, I-PER, O, O, O] |
| SRC | [.., and, ordered, **Lloyd, 's**, to, give, ..] |
| SRC Token | [.., and, ordered, **Lloyd, 's**, to, give, ..] |
| Truth | [.., O, O, I-ORG, I-ORG, O, O, ..] |
| LSTM | [.., O, O, I-ORG, O, O, O, ..] |
| LSTM+CRF | [.., O, O, I-ORG, I-ORG, O, O, ..] |
| SRC | [TENNIS, -, **SPAIN**, ,, U.S., TEAMS, ..] |
| SRC Token | [TENNIS, -, <**unk**>, ,, U.S., <unk>, ..] |
| Truth | [O, O, I-LOC, O, I-LOC, O, ..] |
| LSTM | [O, O, I-PER, O, I-MISC, I-MISC, ..] |
| LSTM+CRF | [O, O, I-LOC, O, I-LOC, O, ..] |
| SRC | [–, New, York, **Newsdesk**, 212-859-1610, .] |
| SRC Token | [–, New, York, <**unk**>, ¡unk¿, .] |
| Truth | [O, I-ORG, I-ORG, I-ORG, O, O] |
| LSTM | [O, I-ORG, I-ORG, I-ORG, O, O] |
| LSTM+CRF | [O, I-LOC, I-LOC, O, O, O] |
| SRC | [Ford, China, **JV**, posts, ..] |
| SRC Token | [Ford, China, <**unk**>, posts, ..] |
| Truth | [I-ORG, I-ORG, O, O, ..] |
| LSTM | [I-ORG, I-ORG, O, O, ..] |
| LSTM+CRF | [I-ORG, I-ORG, I-ORG, O, ..] |

Table 4: Evaluation Set Prediction Samples

how the span-level metrics are performing. Second, I did token level analysis on OOV vs IV word sets.

| | | LSTM | LSTM+CRF |
|---|---|---|---|
| OOV P:10403 | Pr | 68.2% | 73.3% |
| | Re | 68.0% | 69.2% |
| | F1 | **68.1** | **71.2** |
| IV P:767 | Pr | 95.2% | 96.1% |
| | Re | 97.0% | 98.0% |
| | F1 | **96.1** | **97.1** |

Table 5: OOV-IV Span-level Metrics

From Table 5, we can clearly see that the F1 increment is much higher for sentences having OOV tokens. Around 80% of evaluation sentences have OOV which is expected as the CoNLL training dataset is smaller in size. The F1 for OOV present sentences is around 70 where as the F1 for IV sentences is around 96 indicating the challenge in predicting OOV tokens. Comparing the two models across both sentence sets, we can clearly see that F1 improved by >3% for OOV sentences, where as it improved by just 1% for IV sentences. This shows that the CRF is doing a better job in predicting OOV phrases/sentences.

Going into token level predictions in Table 6, we can see that the new model is predicting very close to baseline model for IV token set across all tags. Whereas for the OOV set, the new model is performing better for O, LOC and PER tags, and worse for MISC and ORG tags. Since the contribution of MISC and ORG is lesser, we see an overall improvement in predictions for the OOV set.

| | | Token F1 | |
|---|---|---|---|
| **OOV** | **Tokens** | **L** | **L+CRF** |
| O | 1431 | 84.9 | 85.9 |
| LOC | 105 | 48.3 | 50.0 |
| MISC | 93 | 23.8 | 10.1 |
| ORG | 243 | 61.2 | 54.6 |
| PER | 449 | 75.7 | 80.4 |
| **IV** | **Tokens** | **L** | **L+CRF** |
| O | 7923 | 99.6 | 99.6 |
| LOC | 319 | 92.2 | 95.0 |
| MISC | 171 | 86.4 | 86.5 |
| ORG | 247 | 85.7 | 85.4 |
| PER | 189 | 96.2 | 95.6 |

Table 6: OOV-IV Tag-wise Token-level Evaluation Metrics

| True vs Pred | O | LOC | MISC | ORG | PER |
|---|---|---|---|---|---|
| O | 88% | 1% | 1% | 5% | 5% |
| LOC | **43%** | 41% | 1% | 12% | 3% |
| MISC | **71%** | 5% | 6% | 9% | 9% |
| ORG | **35%** | 2% | 1% | 56% | 6% |
| PER | 11% | 0% | 0% | 6% | 83% |

Table 7: OOV Token-level Confusion Matrix

Table 7 shows the confusion matrix (normalized at row level for each true tag) at token level for OOV tokens using the LSTM CRF model. It shows that almost 40% of the LOC and ORG are misclassified as O, where as this value is 71% for MISC tag. A deep-dive is needed to understand the low performance of MISC and ORG tokens based on the context, neighboring labels etc.

### 3.5 Sentence Length Analysis

One interesting aspect to analyze is whether the new model is able to perform better on longer sentences. Hypothesis is that since CRF does global normalization and takes label dependency (even though it's just adjacent labels for linear-chain CRF), it may help improve the tag predictions for longer sentences. I divided the sentences by their lengths (0-10, 10-20, 20-30, 30+) having (428, 161, 115, 96) sentences respectively. Fig 4 show the span-level F1 scores across these groups for both

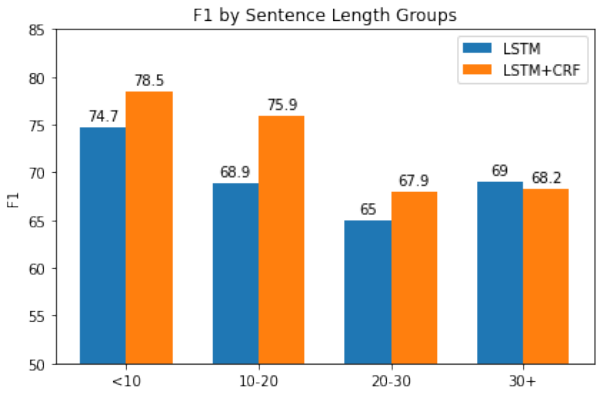


Figure 4: Span F1 by Sentence Length Groups

the models. We can clearly observe that as the sentence length increases, the gains with CRF increases till medium-range lengths, and after 20, the gains drop. CRF performs very well in 10-20 sentence length range giving an impressive F1 increment of 7%. This may suggest that the linear chain CRF works very well for low-mid length range sentences and for extremely longer sentences, we may need higher order CRFs or features that learn long-range tag dependencies.

## 4 Conclusion and Future Work

We have seen that the BiLSTM CRF model improves the NER tagging performance by 2-3% and helps in predicting OOV tokens better. The MISC and ORG class performance are lower in general and there is not much gain with the CRF model as well. The future work will include understanding why and where these classes are inaccurately predicted and use that information to develop features in the model. Also, we can add handcrafted linguistic features, or the character level learnt embeddings along with the LSTM features to improve the performance.