CSE 291E: Project 2

Keerthana Purushotham

I. Abstract

In this project I present and describe my implementations of BiLSTMs with Conditional Random Fields (CRFs) for Named Entity recognition (NER) and compare it to a baseline BiLSTM Tagger for the same task. I analyze their individual performance and contrast them to show how BiLSTMs with CRFs are superior.

II. Named Entity Recognition:

Named Entity Recognition (NER) is the precursor to many NLP tasks that involve context based predictions. Without going into much detail, we can say that it helps classify or rather tag sequences of words in the data into specific categories. The BIO (Beginning, Inside, Outside) tags we use within the context of this project are

['O', 'I-PER', 'I-ORG', 'I-LOC', 'I-MISC', 'B-MISC', 'B-ORG', 'B-LOC', '<start>', '<stop>']

Where 'O' stands for Outside Sequence and is used to tag any word outside the sequence of words representing one entity. 'I-PER' is a tag that represents a word inside the sequence representing a person and so on with 'I-ORG', 'I-LOC' and 'I-MISC' similarly representing tags for words inside sequences representing Organizations, Locations and Miscellaneous things respectively. Again similarly, the 'B' tags represent the Beginning or first word in equivalent sequences.

On a high level view, in order to computationally understand the context of a word in a sentence we need all the information given to us by the word. RNNs are more than capable of computing this. In addition, since languages- specifically English work in a way where in order to make accurate sense of the context of the word (since there are usually many ways the same word can be used grammatically), you need to know the context of the whole sentence and RNNs are incapable of making sense of that much information. Which is why we progress to using LSTMs which are equipped to compute the feature representations of a word in the context of how it is placed in a sentence.

Now, when we look at the specific task of Named Entity Recognition, and how General English Grammar works, we can say that every word in the sentence can convey different things based on how it is placed amongst other words and the sequence in which these words are placed. In other words, the sequence of words placed before and after a given words can by themselves imply certain information about the word itself and we need a way t keep track of this which is why we use Bi-directional LSTMs so we can understand how sequences that come before a word can affect it and the ones that come after it can affect it.

III. Approaches

A. *BiLSTM Taggers:*

As we know RNNs are good tools to compute learned feature representation of sequences and we replace them with LSTMs just to get more context of the rest of the relevant parts of the sentence. The sentence is passed through an embedded layer first and then fed into the bilstm layer. Here there is a forward pass LSTM Layer and the Backward pass LSTM layer which are then concatenated to get the emission probabilities for each word in each sentence in training data. This layer is then linearly mapped through a tag projection layer to the list of tags. These mappings give us the predicted tags.

We evaluate the performance of this model using accuracy, losses and an F1 score using the Conlleval module which essentially calculates the precision, recall and F1 score of the model per epoch.

The F1 score is broken down as follows: First the precision and recall for all our tags are calculated. This is because the accuracy is not an accurate measure of performance on its own because there are so many words outside of named entity sequences that are easily classified to be outside the sequence giving us a deceptively high accuracy. So we use precision and recall for each tag and take the harmonic mean of the two which becomes your F1 score for each tag and the average of F1 scores for all tags becomes your final F1 score.

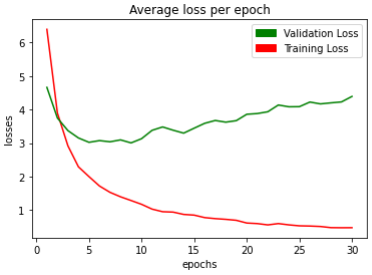

Fig. 1. Training loss and Validation loss over 30 epochs

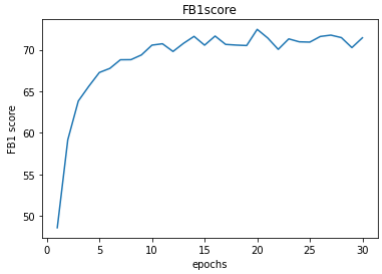

Fig. 2. F1 scores across 30 epochs

```
--- EPOCH 22 ---
Avg loss over last 500 updates: 0.5848652656900732
Avg loss over last 500 updates: 0.7145347758563492
Avg loss over last 500 updates: 0.6144893635343085
Avg loss over last 500 updates: 0.5708239249768722
Avg loss over last 500 updates: 0.44241634180171124
Avg loss over last 500 updates: 0.6518364265900527
Avg evaluation loss: 4.135060921451852
processed 11170 tokens with 1231 phrases; found:
1178 phrases; correct: 859.
accuracy:  75.55%; (non-O)
accuracy:  94.46%; precision:  72.92%; recall:  69.78%;
 FB1:  71.32
LOC: precision:  83.53%; recall:  79.61%; FB1:  81.52  346
MISC: precision:  81.29%; recall:  58.85%; FB1:  68.28  139
ORG: precision:  65.83%; recall:  59.61%; FB1:  62.56  278
PER: precision:  66.02%; recall:  74.25%; FB1:  69.90  415
(72.92020373514431, 69.78066612510155, 71.31589871315899)
```

Fig. 3. Training and Validation output

```
5 random evaluation samples:
SENT: <unk> <unk> gets the opportunity to enter the next
<unk> with such a big <unk> has got to be seen as a
strong message to the industry . "
TRUE: O O O O O O O O O O O O O O O O O O O O O O O O O O O
PRED: O O O O O O O O O O O O O O O O O O O O O O O O O O O
SENT: 2. <unk> <unk> ( Finland ) <unk> <unk> <unk>
TRUE: O I-PER I-PER O I-LOC O I-MISC I-MISC O
PRED: O I-PER I-PER O I-LOC O O O O
SENT: Martin <unk> ( Czech Republic ) beat 0000 - <unk> El <unk>
TRUE: I-PER I-PER O I-LOC I-LOC O O O O I-PER I-PER I-PER
PRED: I-PER I-PER O I-LOC I-LOC O O O O I-PER I-PER I-PER
SENT: 10. <unk> <unk> ( Britain ) 0000
TRUE: O I-PER I-PER O I-LOC O O
PRED: O I-PER I-PER O I-LOC O O
SENT: PRESS DIGEST - <unk> - AUG 0000 .
TRUE: O O O I-LOC O O O O
PRED: O O O I-LOC O O O O
```

Fig. 4. Random Samples

Fig. 1 is a plot of the training loss and the validation loss over 30 epochs and we see that there is some overfitting after around epoch 3. Fig 2 is a plot of the F1 scores across 30 epochs. We see that after reaching a value of about 71 at around epoch 10, it just meanders between 70 and 73 and peaks at 72.46256239600665. This is a parallel reflection of the overfitting and the decrease in change of the loss since it indicates that not much is being learnt by the model after this point. Fig 3 contains the outputs of the training and validation and how it progresses. Fig 4 is some random samples along with their True tags and Predicted tags.

B. BiLSTM_CRF

1) BiLSTM Layer: As mentioned above, the input sentence is fed into the embedded layer and then the BiLSTM Layer where the features of the words are computed in forward and backward directions. This will now instead of just getting mapped to a label based on the calculated probability value, we will further take these probabilities and map them to a conditional random field.

2) CRF: The conditional random field in essence is a transition matrix with values that denote the cost of moving from one tag to another. This transmission matrix is dependent on the grammar of a particular language since it is the language specific grammar that specifies how words in sequences make sense.

3) Transition and Emission Probabilities: Just to bring to the context of this report, Transition probabilities are the probabilities with which one tag given the input word at that time-step maps to the tag of the word of the next time step. Emission probabilities are the probabilities based on which a given word is mapped to a tag.

4) The Viterbi Algorithm: To be short, the Viterbi algorithm finds an optimal path through a sequence given a cost function by tracing backwards through a path of all possible paths.

At every time step, the algorithm looks back one step and eliminates the least optimal edges from the previous tag sequence to the current tag sequence and follows this through till the end of the sentence. Then it backtracks to find the single highest scoring path. The scores are just a simple product of all emission and transition probabilities at each step. This as we know translates into the product of the sum of the feature potentials from the BiLSTM (our emissions) and the transition matrix.

This highest scoring path then becomes the predicted sequence of tags for the given sentence. Just like in the previous approach, we evaluate the performance of this model using accuracy, losses and an F1 score using the Conlleval module which essentially calculates the precision, recall and F1 score of the model per epoch.

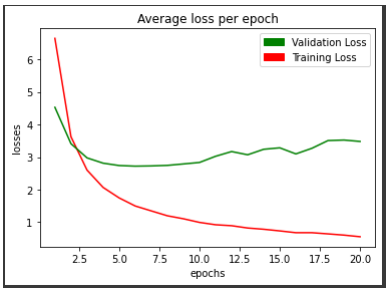

Fig. 5. Training loss and Validation loss over 20 epochs

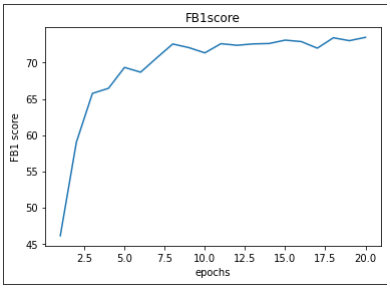

Fig. 6. F1 scores across 20 epochs

Fig. 5 is a plot of the training loss and the validation loss over 20 epochs and we see that there is some overfitting after around epoch 3. Fig 6 is a plot of the F1 scores across 20 epochs. We see that after reaching a value of about 72 at around epoch 7, it just meanders between 70 and 74 and peaks at 73.50210970464134. We can see that the learning somewhat stabilises after the first few epochs. We also see that the general average F1 score is better than that of the previous approach. Fig 3 contains the outputs of the training and validation and how it progresses. Fig 4 is some random samples along with their True tags and Predicted tags.

```
--- EPOCH 19 ---
Avg loss over last 500 updates: 0.5262013514041901
Avg loss over last 500 updates: 0.6730352051258087
Avg loss over last 500 updates: 0.568339765548706
Avg loss over last 500 updates: 0.5067094912528992
Avg loss over last 500 updates: 0.46600664710998535
Avg loss over last 500 updates: 0.5581764268875122
Avg evaluation loss: 3.4788727389276026
processed 11170 tokens with 1231 phrases; found: 1139
phrases; correct: 871.
accuracy:  75.94%; (non-O)
accuracy:  94.66%; precision:  76.47%; recall:  70.76%;
 FB1:  73.50
LOC: precision:  85.50%; recall:  79.61%; FB1:  82.45  338
MISC: precision:  76.87%; recall:  58.85%; FB1:  66.67  147
ORG: precision:  71.59%; recall:  61.56%; FB1:  66.20  264
PER: precision:  71.79%; recall:  75.88%; FB1:  73.78  390
(76.47058823529412, 70.75548334687247, 73.50210970464134)
```

Fig. 7. Training and Validation output

```
5 random evaluation samples:
SENT: But once the <unk> Stich got on the court , he <unk>
his <unk> on trying to win the year 's last Grand Slam .
TRUE: O O O O I-PER O O O O O O O O O O O O O O
O O I-MISC I-MISC O
PRED: O O O O I-PER O O O O O O O O O O O O O O
O O I-MISC I-MISC O
SENT: The Brazilian 's <unk> effort was enough to <unk> Real
 a point from a <unk> 1-1 draw at fellow title contenders
<unk> <unk> .
TRUE: O I-MISC O O O O O O I-ORG O O O O O O O O O O
I-ORG I-ORG O
PRED: O I-MISC O O O O O O O O O O O O O O O O O O O
SENT: <unk> health began to <unk> in 0000 when she was <unk>
 with a heart <unk> .
TRUE: O O O O O O O O O O O O O O O
PRED: O O O O O O O O O O O O O O O
SENT: <unk> 1996-08-29
TRUE: I-LOC O
PRED: I-LOC O
SENT: <unk> 0000 0000 0000 0000 0000 0000 0000
TRUE: I-ORG O O O O O O O
PRED: I-ORG O O O O O O O
```

Fig. 8. Random Samples

IV. Conclusion

The two approaches for NER including just a BiLSTM Tagger and a BiLSTM with CRF have been presented in this report and analyzed to show the performance of each model. Further inferences have been made from the data obtained from the performance analysis and presented above. It is shown that the approach with the CRF is most accurate since in addition to the previous approach also takes into consideration the grammatical context of the language or the dependency between tags to predict tags.

V. References

- https://michhar.github.io/bilstm-crf-this-is-mind-bending/
- Huang, Zhiheng et al. "Bidirectional LSTM-CRF Models for Sequence Tagging." ArXiv abs/1508.01991 (2015): n. Pag.
- Michael Collins. Log-Linear Models, MEMMs, and CRFs.
- Zhiheng Huang, Wei Xu, and Kai Yu. Bidirectional lstm-crf models for sequence tagging. arXivpreprint arXiv:1508.01991. 2015.
- 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.
- Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. Neural architectures for named entity recognition. arXiv preprint arXiv:1603.01360. 2016.
- Xuezhe Ma and Eduard Hovy. End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF. ACL. 2016.
- Erik F. Sang and Fien De Meulder. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. CoNLL. 2003.