Interpreting BERT based Seq2Seq (i.e NER) models

In my teens, I found that cognitive psychology and biotechnology could hold the world in suspended animation. There is a great deal of mystery and joy in understanding the delicate and intricate internal processes that lead to external (visible) action. More than a decade later, pytorch based NLP models turned out be a fair enough substitute. So, here is an analysis of Named Entity Recognition (NER) Model with Captum library.

The model used here is an NER model fine tuned with BERT-base. It was obtained from https://github.com/kamalkraj/BERT-NER

Objectives:

1. Take a target word and visualize how every other word contributes to the final tag for the target word
2. For the target word selected above, zoom into how much information comes word embeddings and position embedding of every word in the sentence
3. Zoom one step further and understand how much each layer of a sample word contributes to the final tag of the target word

We will be exploring how the word "Rob" in the sentence "Soccer - Steve Tiger gets a lucky win , Rob in surprise defeat" gets tagged as "B-PER". We will also look at the effect 12 embedding layers of the word "surprise" have on "Rob" being tagged as "B-PER".

While working with attribution based methods like the ones used in Captum, we will need a reference sentence. In this case, we can substitute every word in the original sentence with '[PAD]' and use it as reference.

```
reference sentence: [CLS] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD]
[PAD] [PAD] [PAD] [PAD] [PAD] [SEP]
```

The algorithm will first see how the information flows from input to the output from baseline. Then it will determine how the information flow differs for the sentence we are interested in.

Effect of words on tagging of target word:

We can create a hook on the model.bert.embeddings layer with LayerIntegratedGradients and obtain the effect of each word on the target word. It can be visualized as follows :

Visualizations For " Rob "

True Label	Predicted Label	Attribution Label	Attribution Score	Word Importance
B-PER	B-PER (1.00)	B-PER	1.11	[CLS] Soccer - Steve Tiger gets a lucky win , Rob in surprise defeat [SEP]

In the image above we see that "surprise" contributed most to tagging "Rob" as "B-PER" while the word "Rob" itself contributed negatively to its tagging. To understand this further, we will look into the contribution of the word embedding and position embedding for the word "Rob" in the next section

Effects of word and position embedding:

To create hooks into multiple layers , we cannot use LayerIntegratedGradients . Instead we will have to use configure_interpretable_embedding_layer to create hooks on bert.embeddings.word_embeddings and bert.embeddings.position_embeddings. After running the forward function of the model through these hooks, we can see the following:

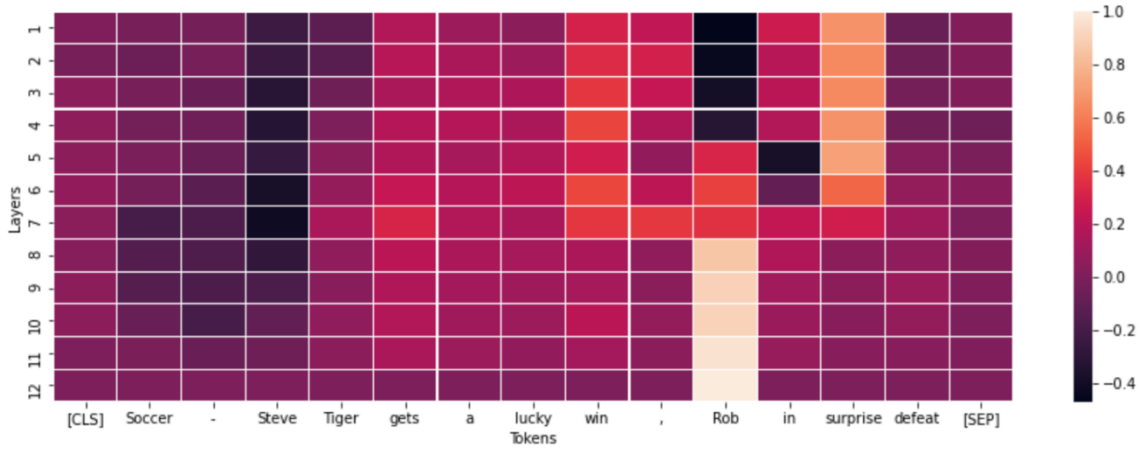
	Word(Index), Attribution	Position(Index), Attribution
0	Rob (10), 0.46	[SEP] (14), 0.73
1	lucky (7), 0.34	[CLS] (0), 0.0
2	surprise (12), 0.34	Steve (3), -0.05
3	gets (5), 0.33	lucky (7), -0.06
4	- (2), 0.33	Tiger (4), -0.06
5	, (9), 0.32	gets (5), -0.07
6	in (11), 0.23	in (11), -0.07
7	a (6), 0.23	win (8), -0.07
8	win (8), 0.22	- (2), -0.08
9	defeat (13), 0.19	a (6), -0.08
10	Tiger (4), 0.15	surprise (12), -0.09
11	Steve (3), 0.14	Soccer (1), -0.1
12	Soccer (1), 0.02	, (9), -0.11
13	[SEP] (14), 0.0	defeat (13), -0.12
14	[CLS] (0), 0.0	Rob (10), -0.63

We find that the word embedding layer for "Rob" contributed positively for it being tagged as 'B-PER'. The overall score for contribution of "Rob" fell down because of it's word position.

Next, we will see how each of the 12 transformer layers of each word in the sentence contributed to tagging "Rob" as a 'B-PER'

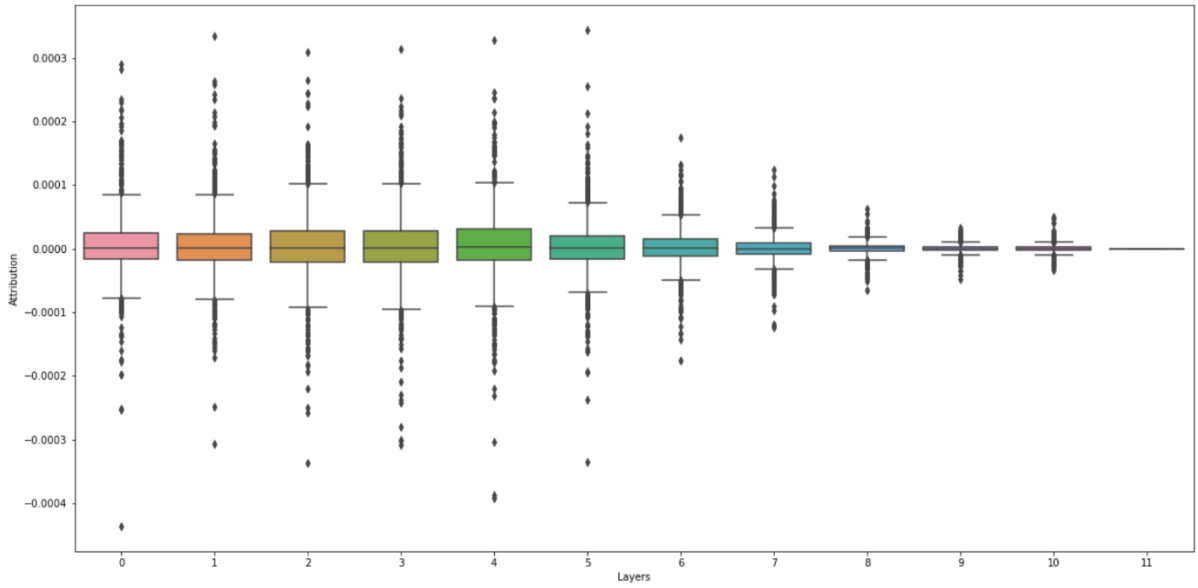
Effects of the 12 transformer Layers:

We once again create a hook on 'bert.embeddings' and use LayerConductance to come up with



From the image above, we can see that the first few layers of the words close to "Rob" have contributed positively to it being tagged as a 'B-PER'. We also find that the last few layers of the target word "Rob" also has a strong positive influence.

In the above table and the visualizations , we saw that the word "surprise" contributed significantly to "Rob" being tagged as 'B-PER'. Lets zoom deeper into each of the 12 layers for the word "surprise"



We see that the attribution values of each embedding the first 6 layers of the word "surprise" is significant. Also, the variance of each embedding in the first 6 layers is high compared to the later layers. The attribution values of embeddings in the later layers are tightly clustered towards zero with little variance.

Code:

A reproducible and commented tutorial jupyter notebook is here (https://github.com/LopezGG/IntrepretingSeq2Seq)