Part 1

Refael Shaked Greenfeld 305030868,

Danit Yshaayahu 312434269

Considerations:

First about word that doesn't appearing the train and do for the test we have a special word "*unknown*" that we apply it on in case such word appears. We also considered to replace in the training rare words with other special word as well in order to get a signal for it, but then we realized that almost 50% of the words appear only once so the combined vector will not represent all of them good.

Regarding the first and last word. We added a special word <s> twice to create a window of size 5 for example: $< s >< s > w_1\ w_2 w_3$ and same for the end of the sentence, by adding 2 special words </s>. It is important to have different words because a word in the beginning or in the end is a different signal for the model.

The training:

We started our tarining with a partially random nuber of 15, for the NER data set. After a few run we saw we actually converge at about 7 without any improvemnet, so form than on we decided ro run solely on **7 epochs**, due to the many hyper parameter tuning involve.

Also in the first part we examine 3 optimizers, SGD, Adam and **AdamW**. In almost all of our expiriments we saw AdamW is the best optimizer so we decided to stick with it. Also we applied early stopping when we save the stage which gave us the best reults on the dev data, and load it in the end in order to avoid overfitting on the trainning data.

NER tagger

From that point on we chose to focus on the following hyper parameters:

- **Hidden dim size**
- **Batch Size**
- **Learning rate.**

The following image is the accuracy on the dev set in the end of the 4th ephch for the Ner where we took our best model from.

Specifacly for the Ner in part 1 our optimizer was Adam with the following hyper parameters:

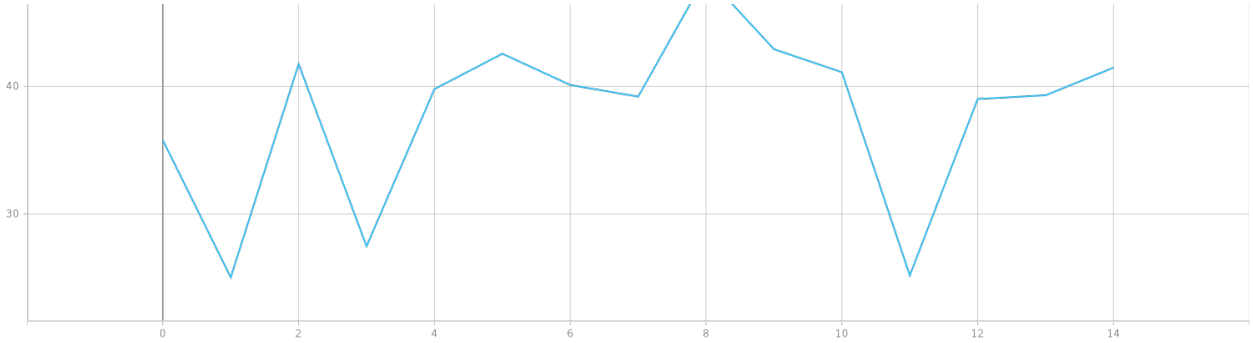
- **Hidden dim size : 200**
- **Batch Size : 3**

- **Learning rate: 0.001**

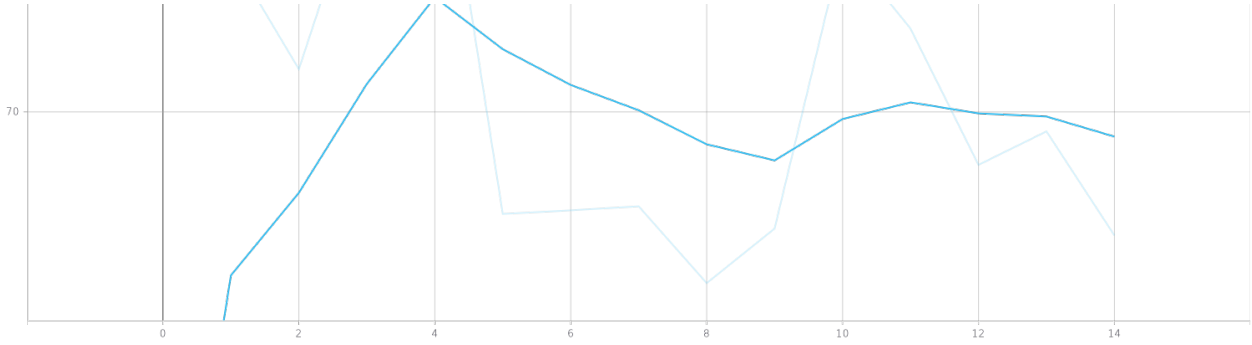
Althogh the loss and accuracy are not aligned in first glance. We need to remmber that the accuracy for the ner is calculated only on the tags that are not 'O' what can cause a slightly disrupted image.

We reached to 78 accuracy in the Ner and 95.1 in the POS in our best runs. Need to mention that the 78 was a very rare run, when it afterwords dropped dramatcly to 67, we savedthe best result on the Dev set, but we can telll that in avrage the model reached 72 % Accuracy and was very bumpy along the tarining.

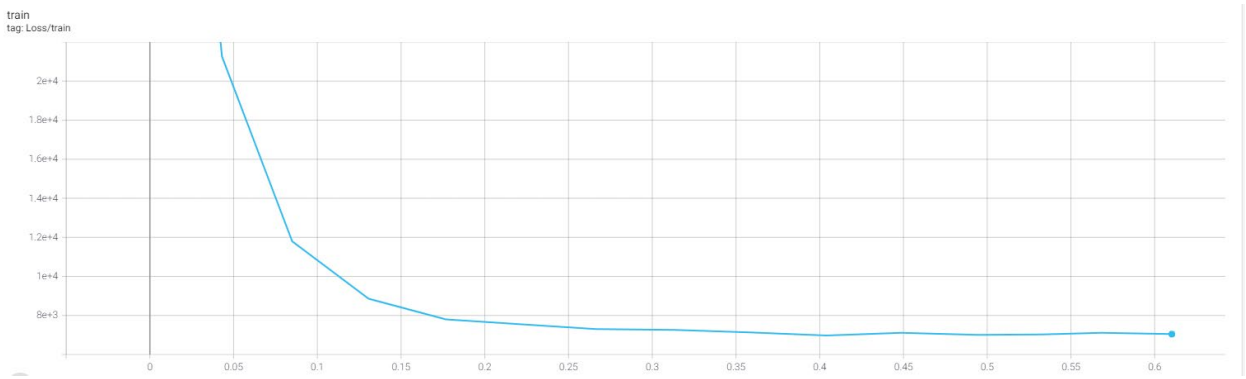
The Ner Loss - Dev:



The Ner accurac - Dev:



We can see that the model it self indeed converged when we look at the **train loss**



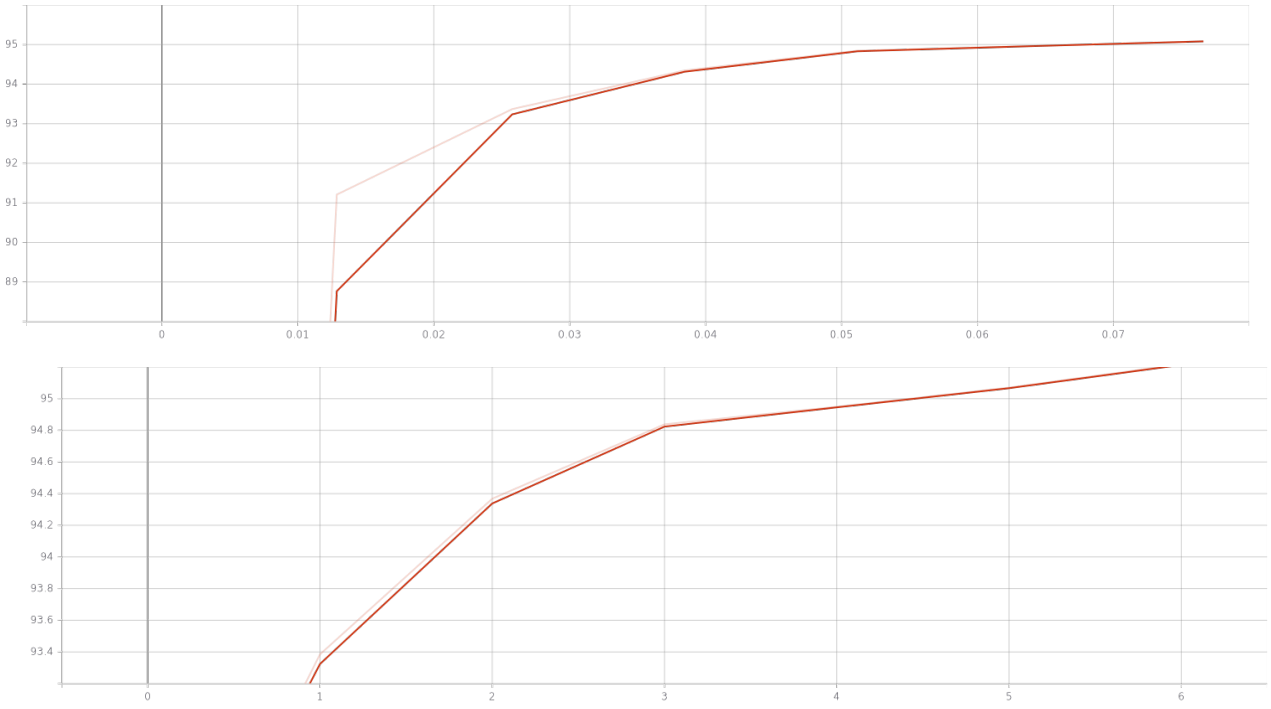
Pos tagger:

As mentioned before we ran with the same optimizer (**AdamW**) for only 7 epochs (and took the best run) the parameters that yield the best results were:

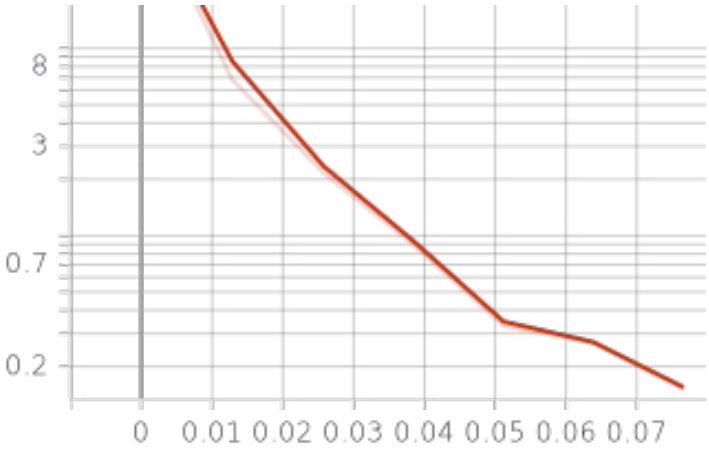
- **Hidden dim size : 200**
- **Batch Size : 128**
- **Learning rate: 0.001**

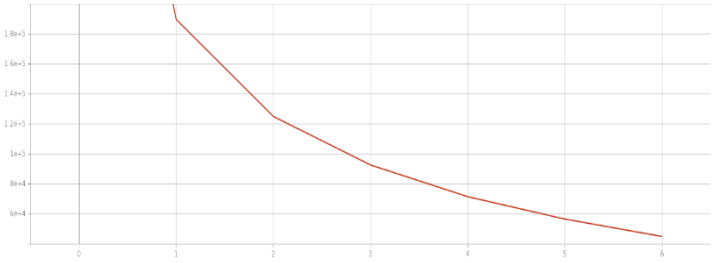
This time the accuracy on the dev and the loss aligned nicely (I added the same garph twice to show it better, the real values are the line which is a bit transparent and the bold one is the smoothed.)

Accuracy Dev set:



Loss Dev Set:





Appendix

A nice graph that show the various dev accuracies of the different runs: