Part 1

Refael Shaked Greenfeld 305030868,

Danit Yishaayahu

## 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. In the training we replace rare words with the “unknown” special word as well in order to get a signal for it (otherwise the “special” vector would actually be random because the model never saw it in the training process).

Regarding the first and last word. We added a special word <s> twice to create a window of size 5 for example and same for the end of the sentence, by adding 2 special words </s>. It is imporatand to have different words because a words in a 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 with out 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 stoppin 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:

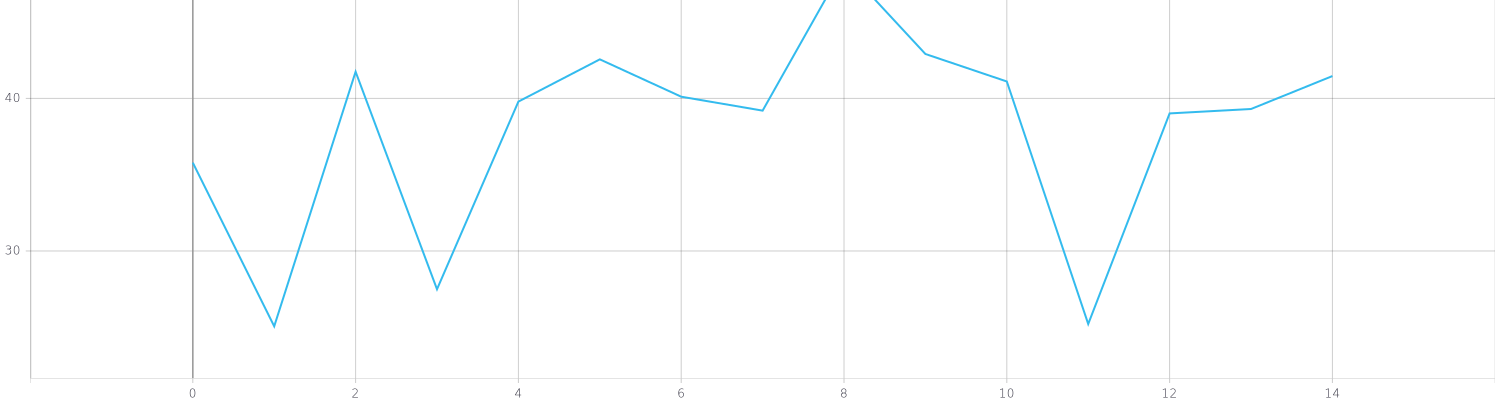
* **Hidden dim size**
* **Batch Size**
* **Learning rate.**

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.

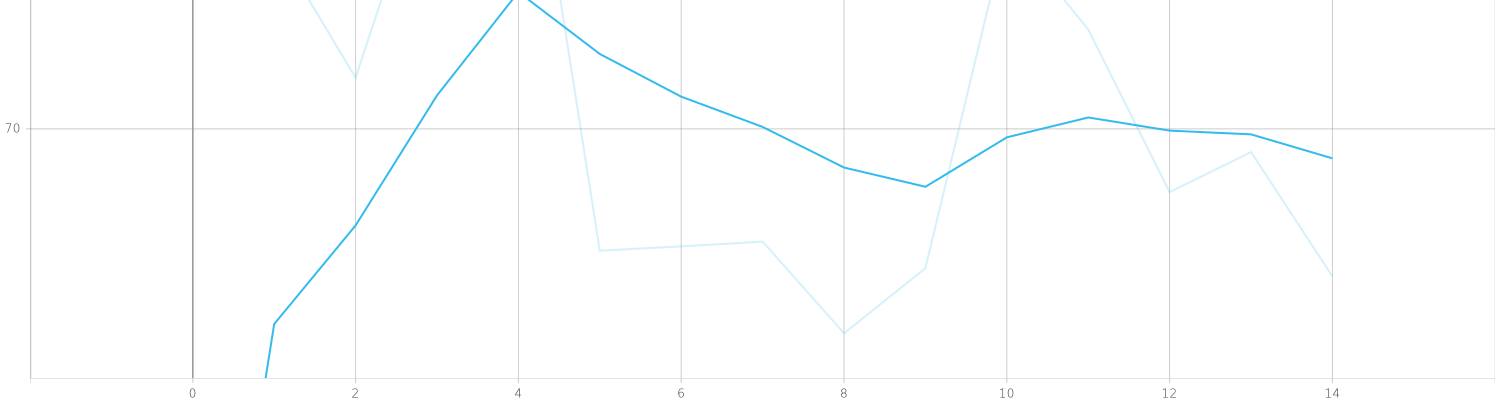
Specifacly for the Ner in part 1 our optimizer was Adam with the following hyper parameters:

* **Hidden dim size : 200**
* **Batch Size : 3**
* **Learning rate: 0.001**

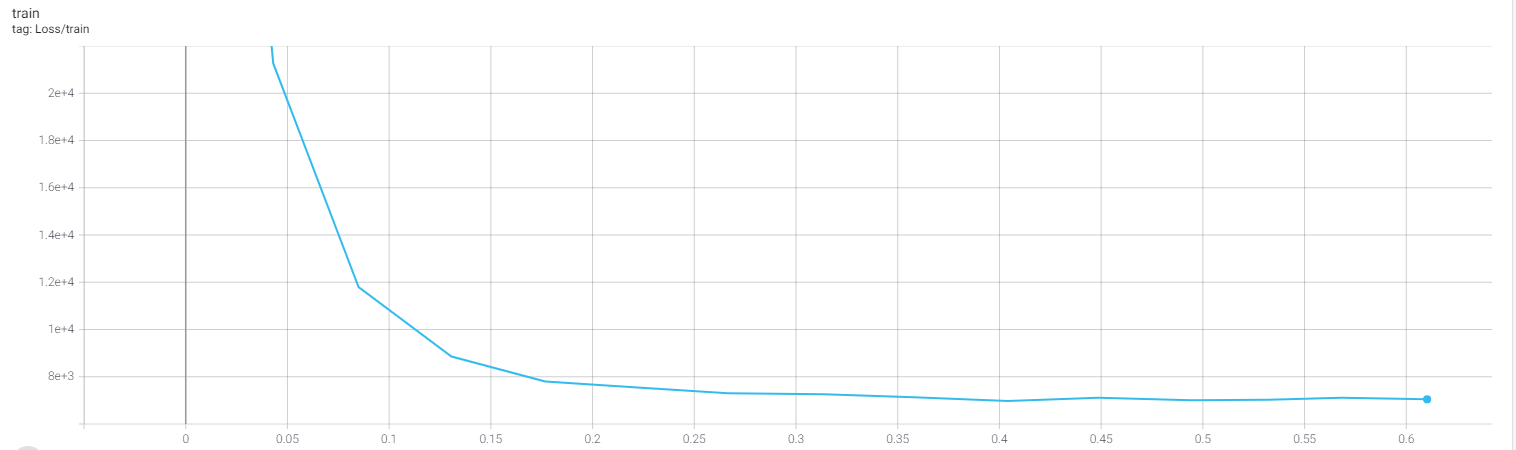
Althogh the loss and accuracy are not allighned in first glance. We need to rembbebr that the accuracy for thr ner is calcilated only on the tags that are not ‘O’ what can cause a slightly ditrupted image.

**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**

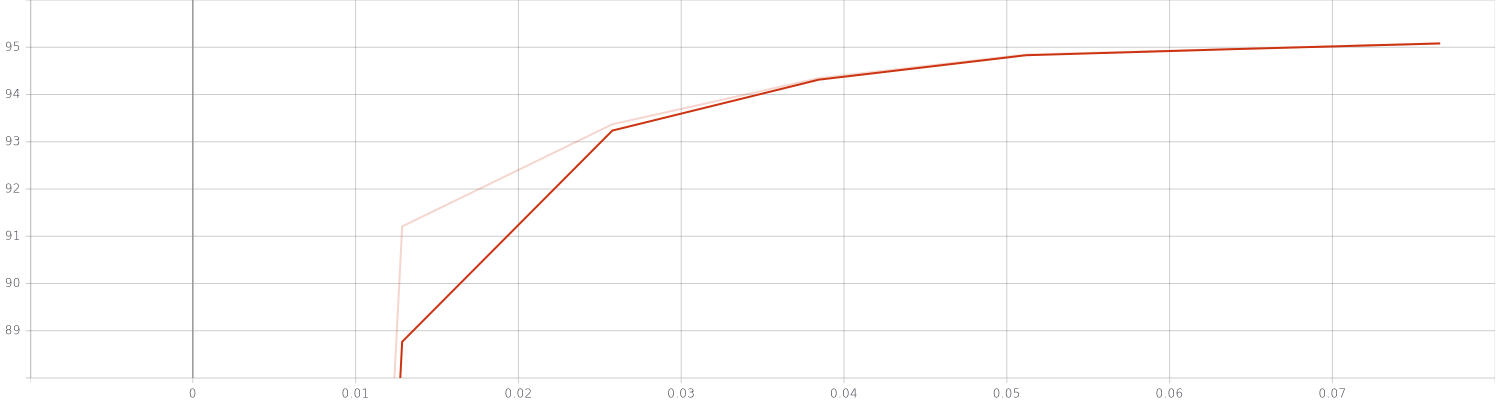
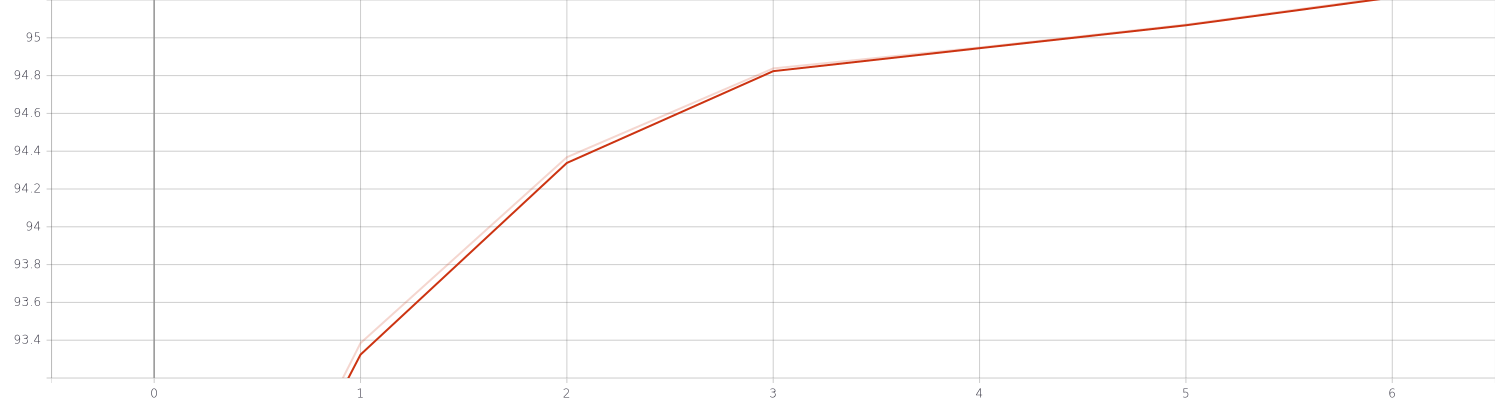


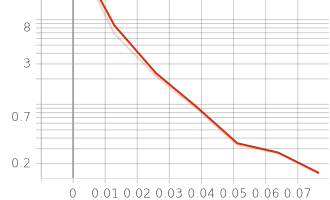
## Pos tagger:

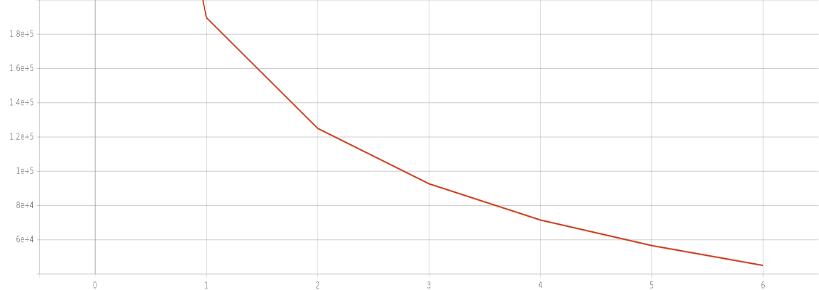
As mentioned before we ran with the same optimizer (**AdamW**) for only 7 epochs (and took the best run) the parameters that yield the best results were:

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

This time the accuracy on the dev and the loss allighned nicely (I added the same garph twice to show it better, the rael values are the ine which are abit transppernd and the bold one are the smoothed.)

Accuracy Dev set:

Loss Dev Set:



## Appendix

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

