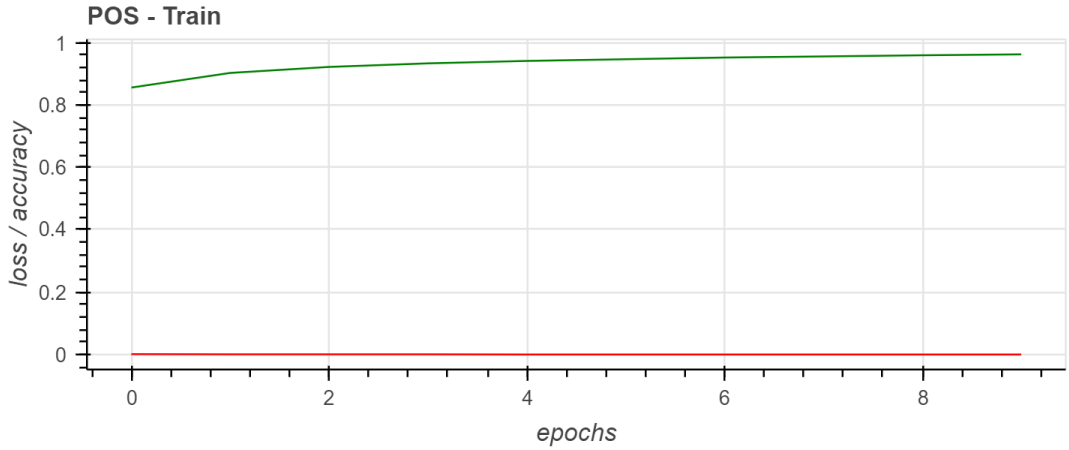
Dafna Magid 315635870 Oved Nagar 302824875

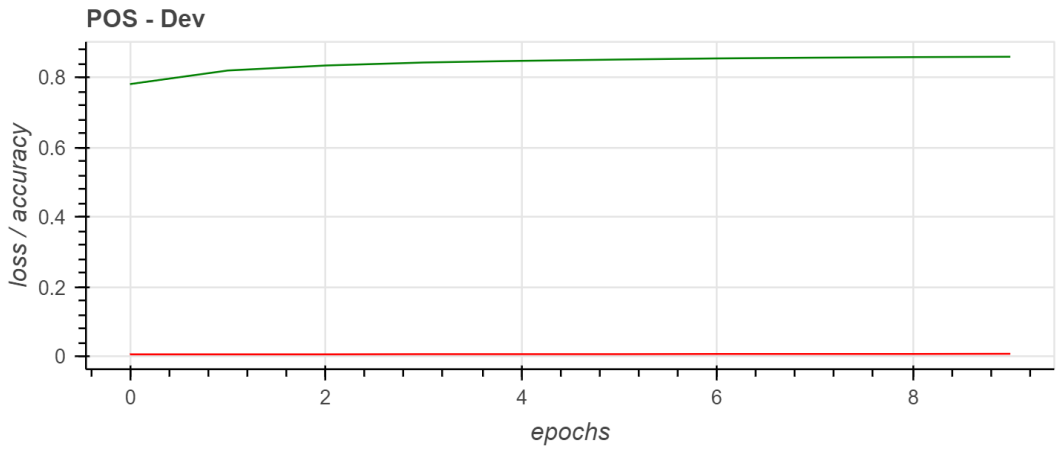
The parameters of the best model for each of the tasks (NER and POS):

	Model POS	**Model NER**
Optimizer	torch.optim.RMSprop	torch.optim.RMSprop
Batch size	512	512
Hidden dim	80	165
Learning rate	0.005	0.005
Number of epochs	10	20
Last epoch accuracy dev	86%	80%

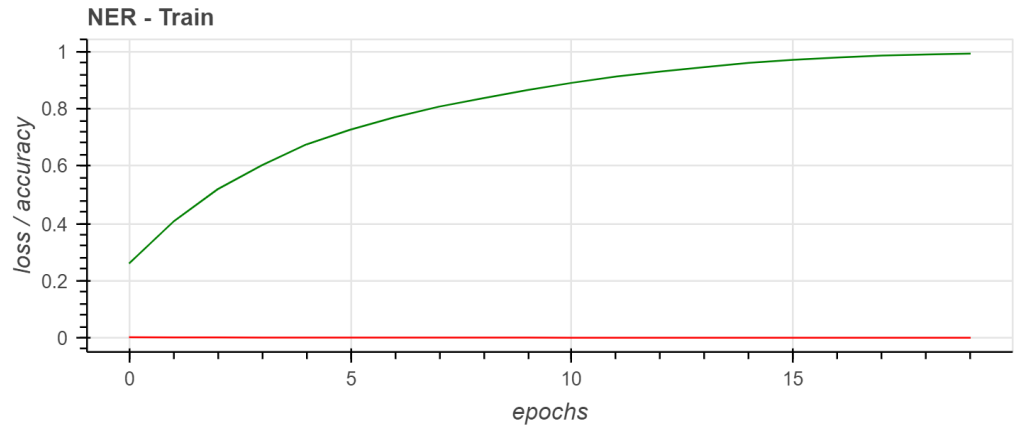
The graphs showing the **accuracy** of the dev set and the **loss** on the dev set as a function of the number of iterations:

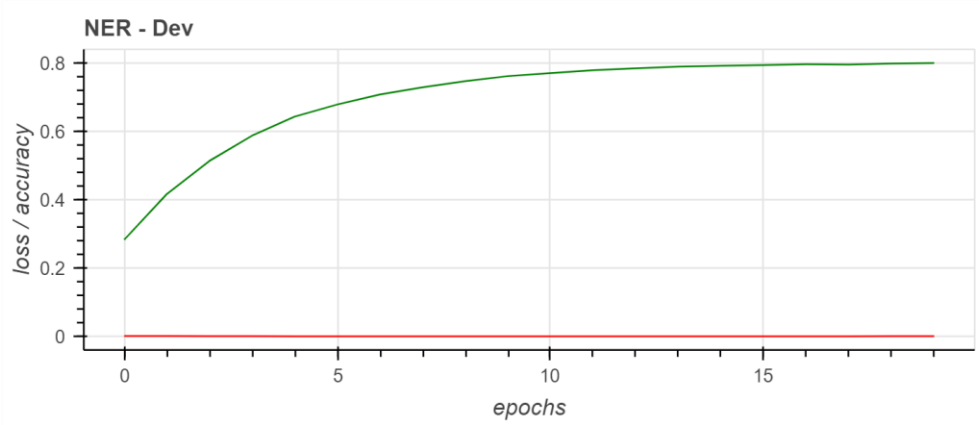
POS-





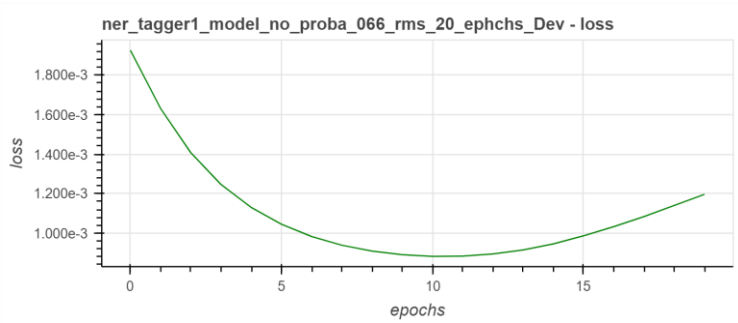
NER-





Accuracy calculation does not include the 'O' label at all!

The loss line may seem linear in the following graphs, but it's not.
For example, for the NER-Dev: the model learns until reaching epoch 10 and then it starts to over fit.



Our solutions to the "considerations":

what do you do with a word that appears in the dev set and not in the train set? which word embeddings will you assign to it?

Assuming that we got a word that appears in the dev set and not in the train set, we do not know her. so, we will look at the words that surround her, two from each direction, and with this information we look for a word in the train set that has the

highest probability to be similar to that unknown word. we will use the representation of the most similar word for our word. There for, the word will also get the same word embeddings representation.

What vectors will you use for the words surrounding the first word in the sequence? And for the last word?

Our data is a long vector of words. Each sentence begins with a mark indicating the beginning of a sentence and ends with a mark indicating the ending of a sentence. When we create a vector for a word, we take the 2 words that surround it from each direction. In the case that the position of the word is in the edges of the sentence, the beginning/end of the sentence marks will also be taken, for which we will not attempt to predict the label Although they also have a representative vector in embedding matrix.