Assignment 2

Part 1

Course: Deep learning for texts and sequences.

Students: Lev Levin, id: 342480456

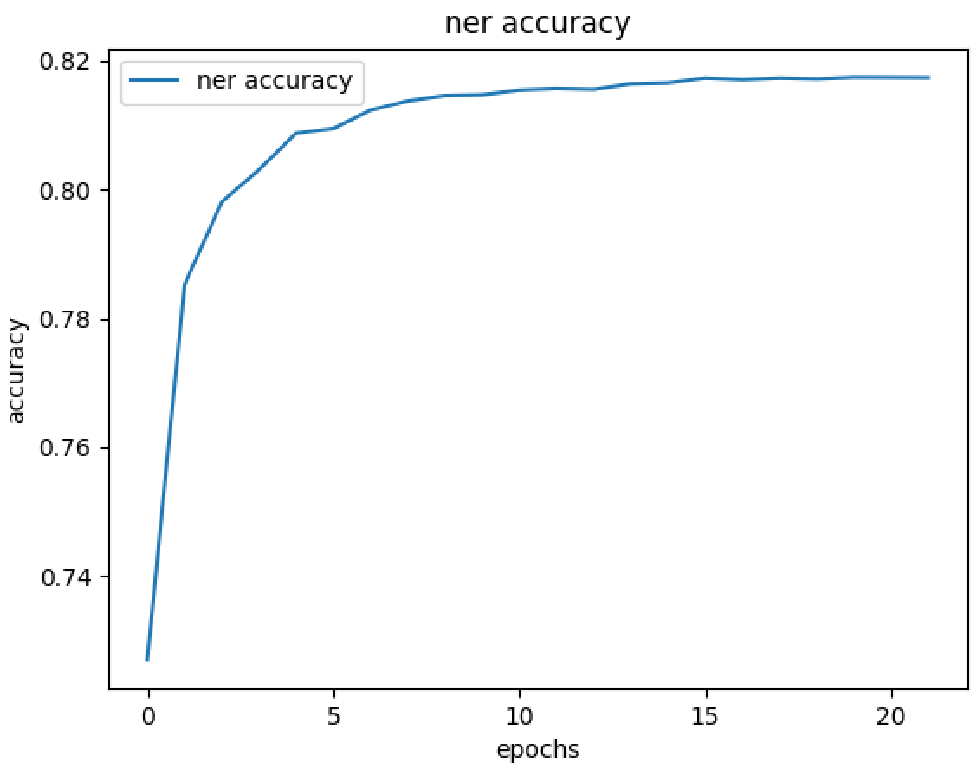
Lior Shimon, id: 341348498

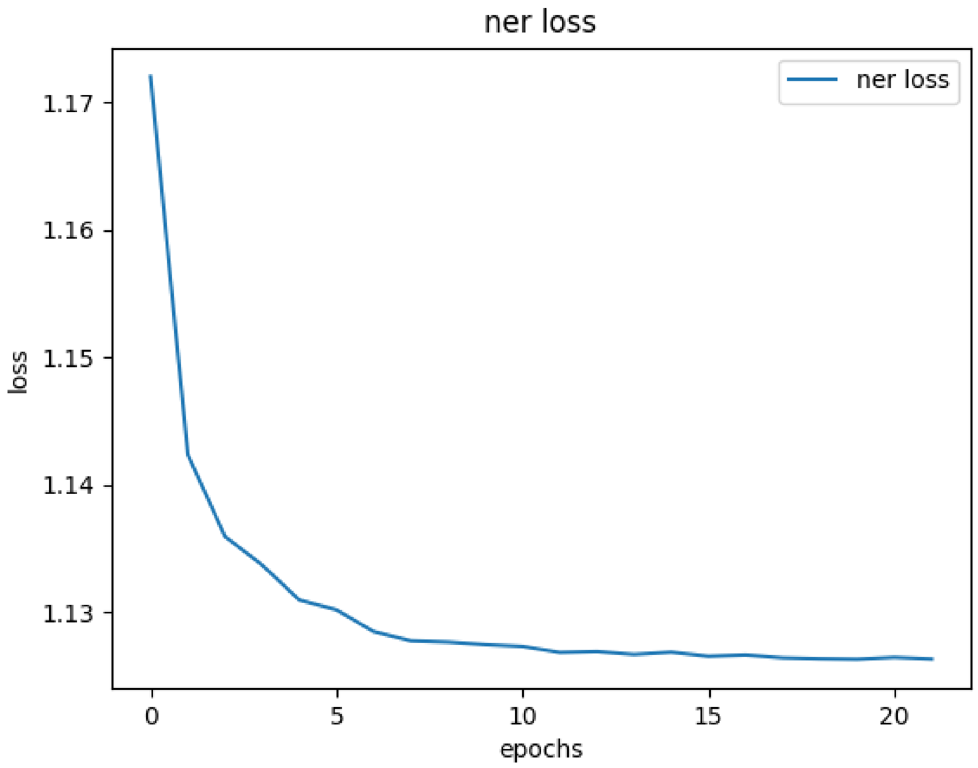
Parameters for ner model:

Description:

```
epochs = 22
hidden_size = 10
l_r = 0.01
torch.manual_seed(1)
batch_size = 1000
Optimizer: Adam
Initialization of embedding: uniform distribution (-1,1)
```

Results:





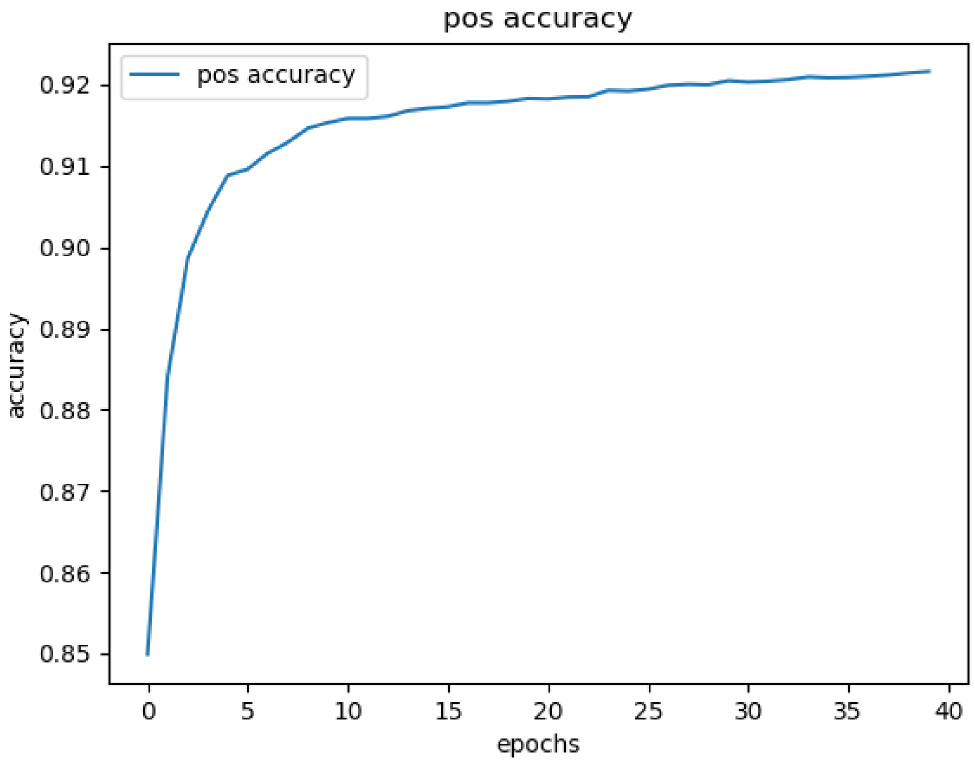
```
Epoch [22/22], Train_Loss: 1.0535, Train_Accuracy: 97.50%
Epoch [22/22] ,Dev_Loss:1.1263,  Dev_Accuracy:81.74%
```

Parameters for pos model:

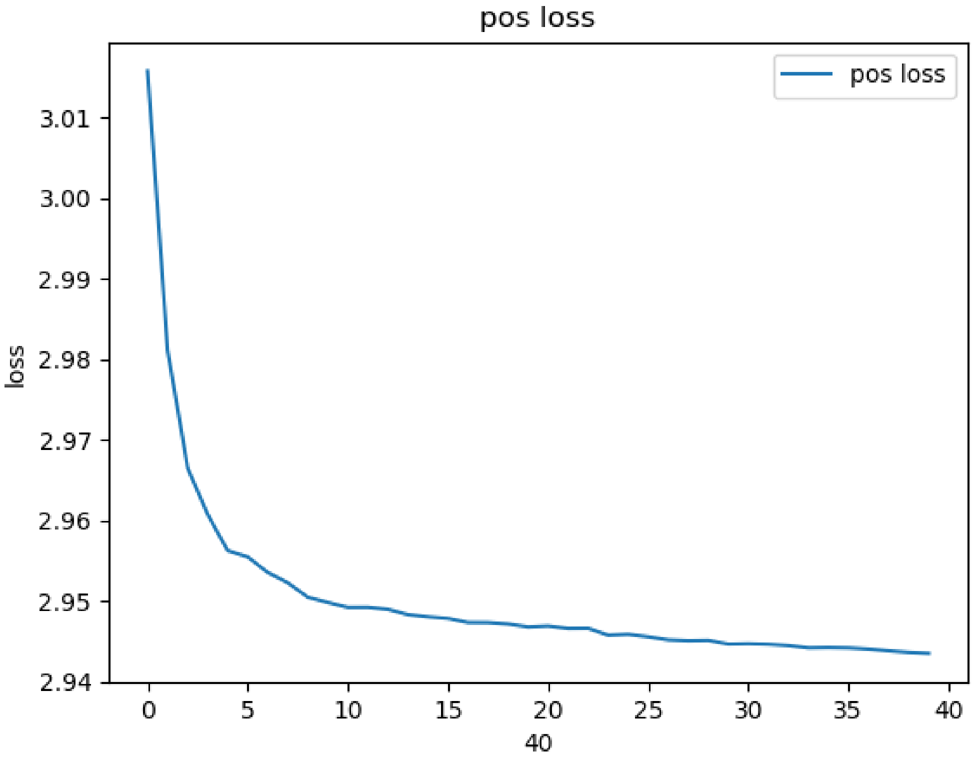
```
epochs = 40
hidden_size = 60
l_r = 0.01
torch.manual_seed(1)
batch_size = 1000
Optimizer: Adam
Initialization of embedding: uniform distribution (-1,1)
```

Results:

Graph of validation accuracy as a function of a number of epochs:



Graph of a loss function over a number of epochs:



```
Epoch [40/40], Train_Loss: 2.9165, Train_Accuracy: 94.86%
Epoch [40/40] ,Dev_Loss:2.9435,  Dev_Accuracy:92.16%
```

Our solutions to 'considerations':

Words that appear in train set but not in dev set:

It doesn’t impact performance of our model because every
word in the dev set makes problem only if it doesn't appear in train set.

Words that appear in a dev set but not in a train set:

While reading sentences from a file, if we get a word that doesn't appear in our main vocabulary which is based on train set, we change the provided word to word '<Miss>' which means – any word that doesn't appear in the vocabulary. We add the '<Miss>' while constructing the vocabulary and initialize it for a vector in embedding matrix. Thus, each word that appears in dev/test set and doesn't appear in train set, we map to the vector '<Miss>'.

After analyzing the model for a while, we decided to ignore tags that were provided for words that doesn't appear in the vocabulary, and to add new label which represents '<Miss>'. It's important to pay attention that while predicting, on some word which is 'miss', we don't take that label as prediction. Instead, we take the label that got the second highest probability after '<Miss>' label. That is, we take a label with highest probability only among part-of-speech labels. Why we decided to create a new label? Because if "miss" will get different labels (and this happens while reading the dev file) we create noise. But if <"Miss"> gets its own label – it indicates that the word is not in the vocabulary and it helps to the Network to learn it and be more precise (as we concluded in empirical way while running our model implementing two different options).

Vectors that we use for the words surrounding the first word:

We added to our vocab a word "<Start>" and initialized for it its own vector in embedding matrix. Each time we want to predict first or second word in the sentence, we put "<Start>" accordingly so that the network would get the input vector in size of 5. For example:

"<Start> <Start> I am going " or "<Start> I am going to"

Vectors that we use for the words surrounding the last word:

We added to our vocab a word "<End>" and initialized for it its own vector in embedding matrix. Each time we want to predict the fourth or fifth word in the sentence, we put "<End>" accordingly so that the network would get the input vector in size of 5. For example:

"long been concerned <End> <End>" or "long been concerned about <End>".