Part 3 Answers

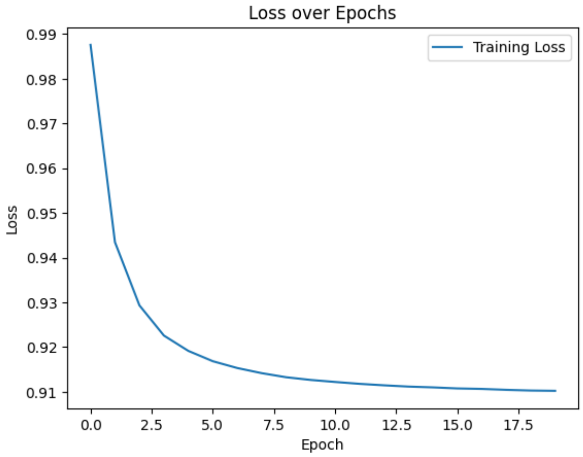
Roee esquira, ID 309840791
Yedidia Kfir, ID 209365188

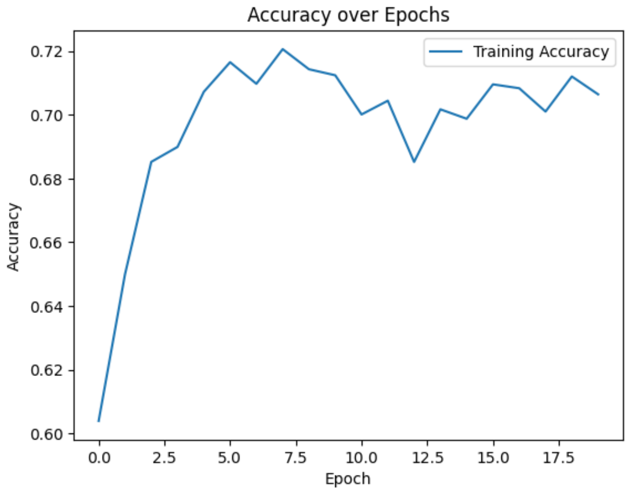
Considerations

1. For words that are found capitalized in train set (i.e Loss or LOSS vs loss), we tried two methods:
    a. The first method was fine tuning the embeddings, and using different vectors for differently capitalized words. Meaning, we used different vectors for Loss and LOSS, but both were initilized for the vector of ‘loss’ in the vectors given. This method did not work well since we could not use the given embeddings for words that are found in dev/test but not found in train, because all the other vectors were finetuned and this vector was not.
    b. The second method, that ended up chosing, was simply ignoring the lower case and using the embedding as is, without fine tuning. We asses that the information in the lower cased embedding is relevant for differently capitalized words.
2. For words that are not found in the pre trained vocabulary, we created new, glorot initilized vectors that we fine tuned in the training process. Notice we only fine tuned these new vectors, and not the pre trained ones.

NER Results

The best parameters we found was using hidden dimension 40 and 0.001 learning rate. This yielded 0.7206000229016375 accuracy rate on dev set after 7 epochs. This is an improvement of 1% accuracy from the tagger without the pretrained embeddings. See accuracy and loss function with hidden dimension 40:



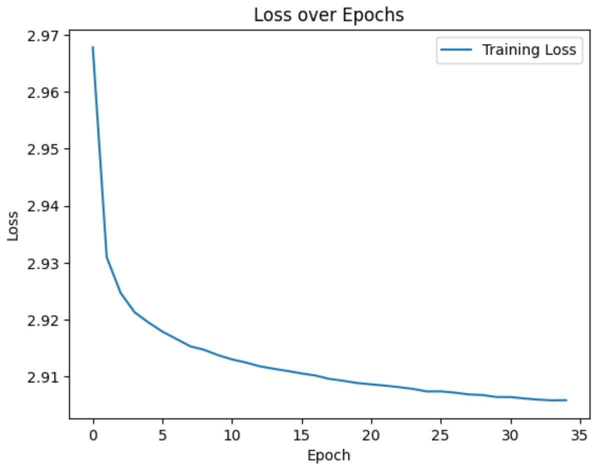


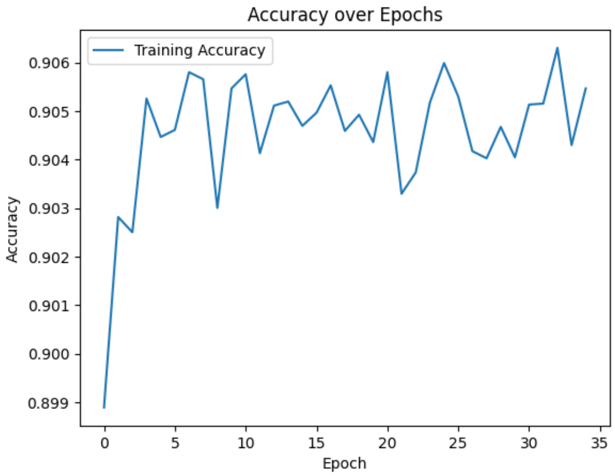
*notice this is accuracy on dev not on train as written in the graph

We also tested with hidden dimension 20, this resulted in peak accuracy of 0.7113201055166877 and hidden dimension 60 that peaked in accuracy of 0. 6993471538197228.

POS Results

The best parameters we found was using hidden dimension 60 and 0.001 learning rate. This yielded 0.9063041617072815 accuracy rate on dev set after 32 epochs. This is an decrease of 2% accuracy from the tagger without the pretrained embeddings. See accuracy and loss function with hidden dimension 60:





*notice this is accuracy on dev not on train as written in the graph

We also tested with hidden dimension 20, this resulted in peak accuracy of 0.8904132473010503 and hidden dimension 60 that peaked in accuracy of 0.9050512643822172.