Part 4 :

Our model includes batch normalization – 100 inputs on each batch .

The size of the hidden layer is 100

Regularization : 5e-5

Optimizer : Adam (pytorch)

Number of epochs – 2 for pos , 6 for ner

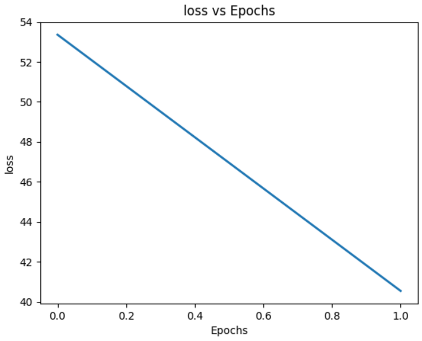
Lr = 1e-3

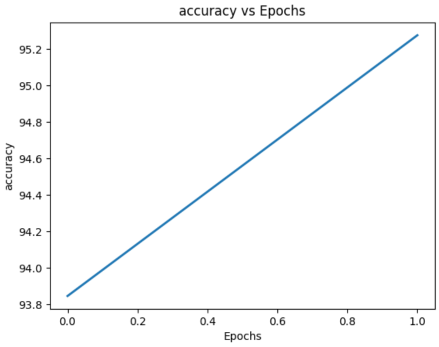
Batch size = 159

We implemented the embedding matrix mechanism like task 3 .

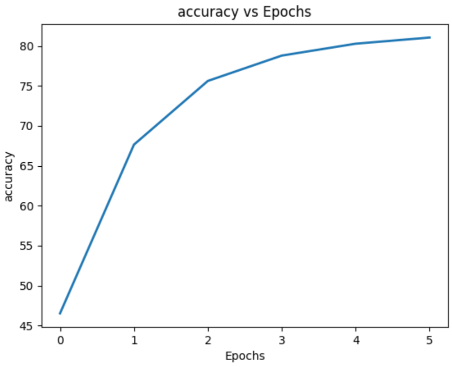
Our results :

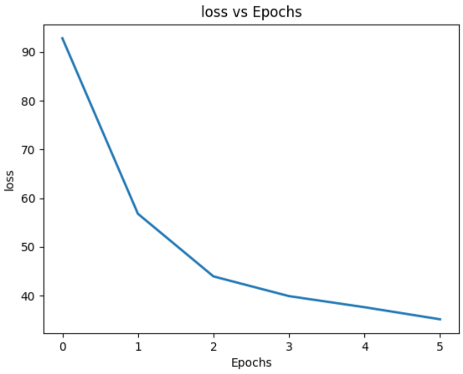
1. Sub-words unit , not pre trained
   Pos : accuracy – 95.27





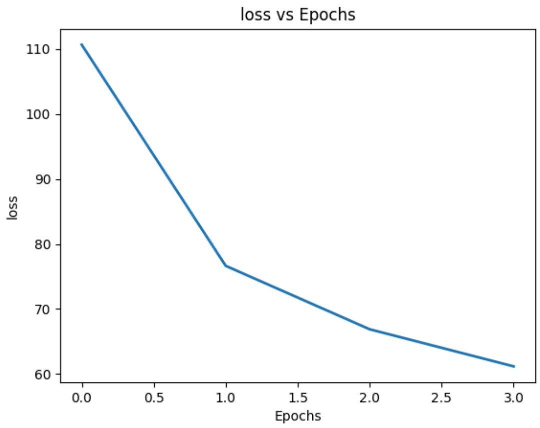
Ner : accuracy – 81.046

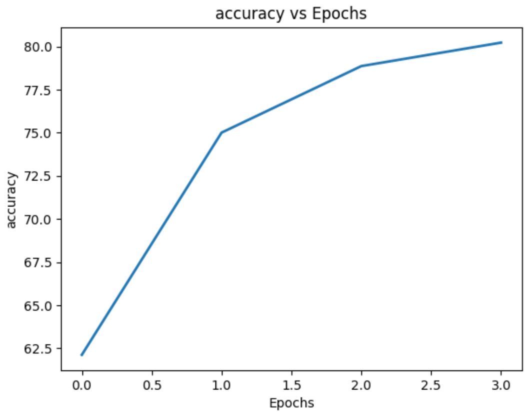




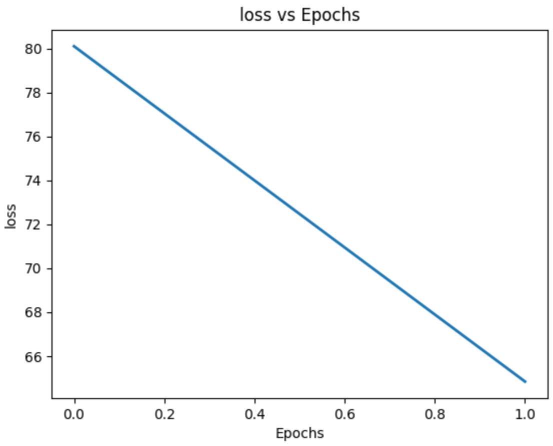
2. Sub – words unit, pre trained

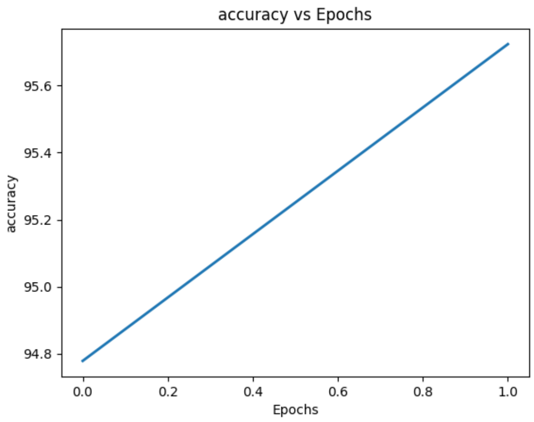
Ner : accuracy – 80.227 4 epochs





Pos : accuracy :95.724 2 epochs





Comparisons

- The model without the pre trained matrix give the same results as the model with the pre trained
- The model with the pre trained and the subwords is better than the model without the subwords .
  The subwords give significant improvement

- We can see that the models with the pre trained matrix aren’t much better
- Conclusion : the models with the subwords have more power than the others .
    - The models with the pre trained matrix aren’t much better then the models without.
    - we think the subwords give significant improvement because there is strong connection between the tagging tasks (pos and ner) and suffixes and prefixes of words. That’s make differences