

Assignment 1 – Part 4

Submitters:

Ori Braverman 318917010

Elie nedjar 336140116

Model Hyper-parameters

- Optimizer: Adam
- Learning rate: 0.001
- Size of hidden layer: 128

In the pos task:

- The batch size of the training was 128 for the training set and the dev set
- The number of epochs was 10 without early stopping but with checkpoint saving when best accuracy is reached.

In the ner task:

- The batch size of the training was 128 for the training set and the dev set
- The number of epochs was 5 without early stopping but with checkpoint saving when best accuracy is reached.

How we handled the pre-trained vectors

- The pre-trained embedding vocabulary is in lower-case. So, when making a match between the words from the train/dev/test to the vocabulary, we first checked for the original word in case it is in the vocabulary and the lower case. This approach helps when dealing with words that have capital letters (e.g. start of a sequence).
- Words recognized as numbers by appropriate regex patterns are assigned tokens from the pre-trained vocabulary that represent numbers, such as 'DGDG', 'DG.DGDG', and 'NNNUMMM'.
- Words in the training, dev, or test sets that do not appear in the pre-trained vocabulary are assigned the unknown token <UNK>. This token is part of the pre-trained vocabulary and represents out-of-vocabulary words.
- For prefixes and suffixes we trained a new embedding for each.

Model Performance

In compared to the previous taggers the results were slightly lower.

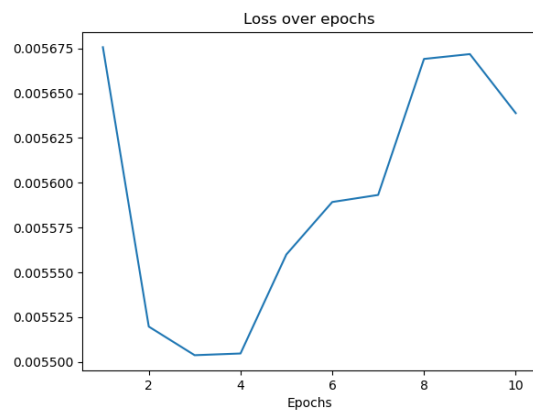
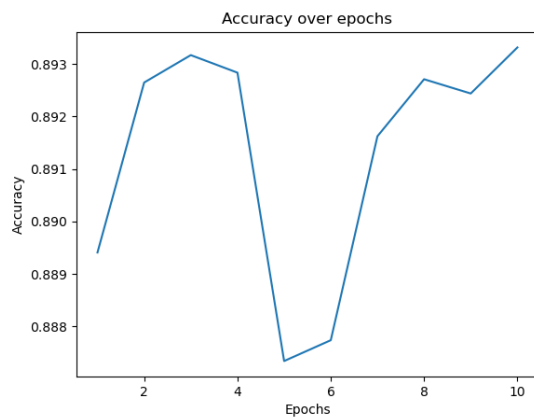
- It can be explained that aggregating multiple vectors by summing them can cause a loss of information, since the embedding of entire words and the ones of prefixes and suffixes are independent

- A solution could have been to concatenate the vectors instead of summing them

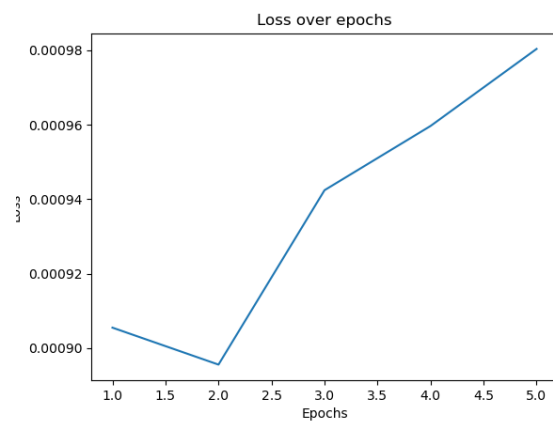
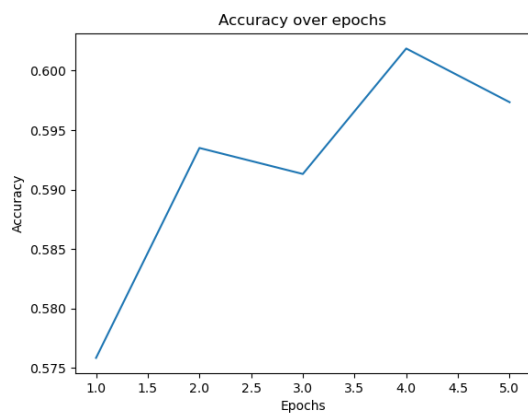
Model Results

With pretrained vectors

POS: - Dev Accuracy: 0.9498 - Dev Loss: 0.0106

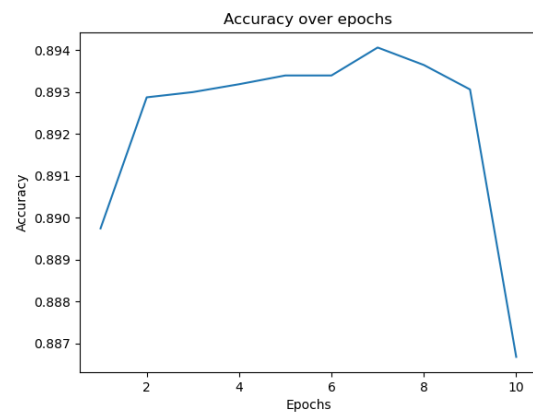
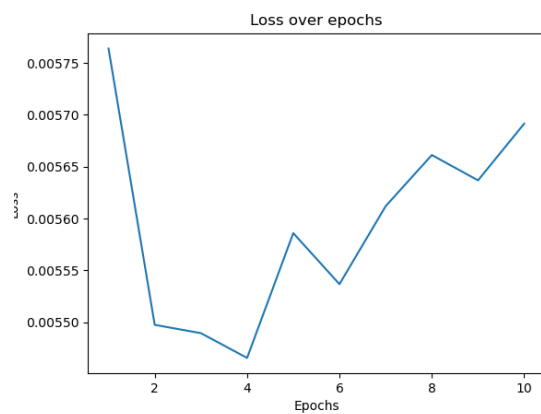


NER: - Dev Accuracy: 0.7671 - Dev Loss: 0.0048



With Random vectors

POS : accuracy 0.894



NER : accuracy 0.57

