

Assignment 2 – Window-based Tagging

Daniel Bazar 314708181

Peleg shefi 316523638

Part 3 – Adding the external word embeddings to the tagger

Architecture

- Both tasks implementing the same Network: a MLP with one hidden layer and a tanh activation function
- The network trained with a cross-entropy loss
- We Experimented with several network configurations and chose the best configuration based on the DEV accuracy

Best parameters

- NER:
 - Hidden layer size: 130
 - Dropout probability: 0.4
 - Batch size: 64
 - Optimizer: Adam (Learning rate: 1e-3)
 - Epochs: 5
- POS:
 - Hidden layer size: 130
 - Dropout probability: 0.2
 - Batch size: 2048
 - Optimizer: Adam (Learning rate: 1e-3)
 - Epochs: 8

Considerations

- We handle words that appear in the training file but not in the embedding file as we handle those words in part1 (words that don't appear in dev file), we assign them the UNK token.
- If we used the pre-trained embedding vectors, we transform the training and the dev data to lower case because the embedding vocabulary being lower-case.
- Because the embedding vocabulary contains special words like "DGDGDGDG", "DG.DG", "+DG", "NNNUMMM", etc. we treated those words as digits patterns.
- We padded the sentences with SOS (start of string) and EOS (end of string) at the beginning and end of the sentence.

Results

Accuracy comparison: The accuracy didn't improve over the tagger without the pre-trained embeddings.

In the NER mission The accuracy get worser and in the POS mission there is no significant change.

- NER:
 - Loss validation: 0.194
 - Accuracy validation: 79.34%
- POS:
 - Loss validation: 0.161
 - Accuracy: 95.1%

Graphs

