Part 4 report:

Tamar Michelson 323805861

Shilo Avital 206487407

The parameters of the best model for both of the tasks (NER and POS):

Hidden dim = 100
Batch size = 32
Num epochs = 10
Learning rate = 1e-3
Weight decay = 1e-5

Part 4 – Subword Units

Model Configuration

In this section, we investigate the effect of integrating subword information—namely prefix and suffix embeddings—into our window-based sequence tagger for both the NER and POS tasks. We trained models both with and without pre-trained word embeddings, allowing us to assess the impact of subword units in isolation and in conjunction with external word knowledge.

Subword Unit Integration

We extracted all 3-character prefixes and suffixes from the training corpus and assigned each a unique embedding vector. During training, the final representation for each word was computed as the sum of its word embedding, prefix embedding, and suffix embedding. These enriched representations were then concatenated across the context window and passed into the MLP classifier.

This approach enables the model to generalize better to rare and unseen words by capturing character-level morphological patterns that are often associated with specific syntactic or semantic properties. For example, the suffix “ing” frequently indicates a present participle verb, while prefixes like “pre” or “un” carry semantic cues.

Use of Pre-trained Embeddings

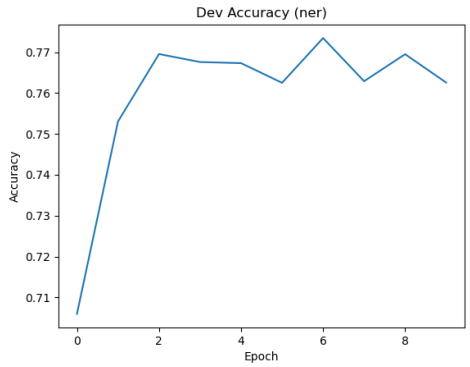
All words were converted to lowercase during preprocessing to ensure compatibility between the dataset and the pre-trained embeddings. Initially, we

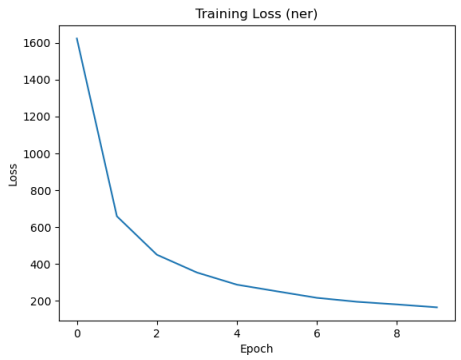
populated the embedding matrix with values sampled uniformly from the range $(\left(-\frac{1}{\sqrt{d}}, \frac{1}{\sqrt{d}}\right), d = embedding_\dim)$where ddd is the embedding dimension. For words that appeared in the pre-trained embeddings file, we replaced their random initialization with the corresponding pre-trained vector. This allowed us to leverage external knowledge while still supporting out-of-vocabulary tokens with learned representations.

Results and Analysis

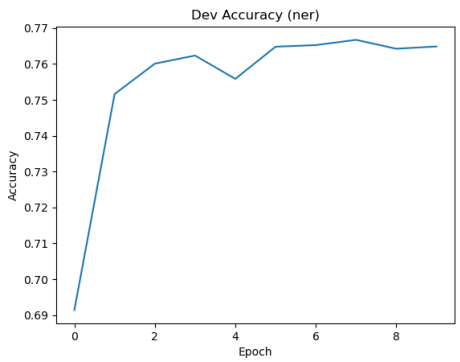
Ner results:

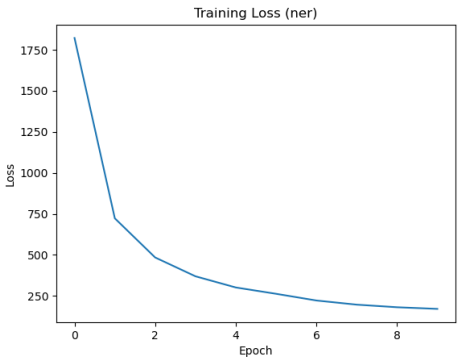
1. **Pretrain embedding :**





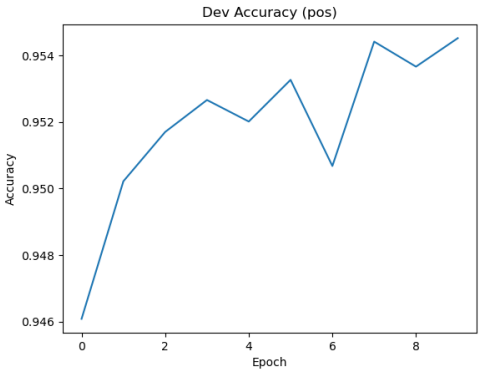
2. **No pretrain embedding :**

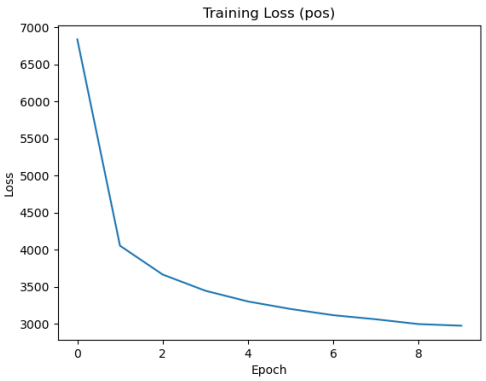




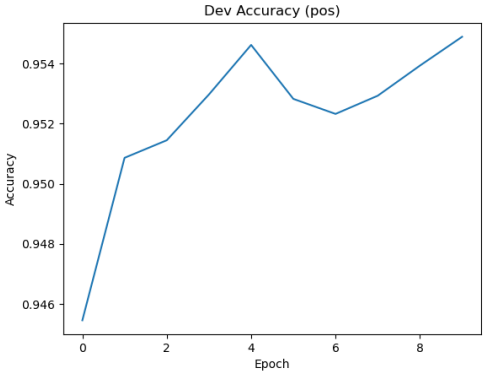
Pos results:

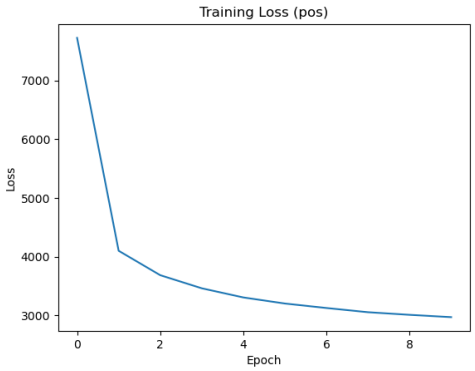
1. **Pretrain embedding :**





1. **No pretrain embedding :**





We present four plots per task: accuracy on the development set as a function of training epochs, and loss on the development set, for both subword configurations—with and without pre-trained embeddings.

Across pos tasks, we observed that integrating subword units consistently improved model performance.

Adding pre-trained embeddings didn't further enhance the model's performance. The pre-trained vectors primarily contributed during the initial training stages, after which the model adapted the embeddings effectively through learning.

Conclusion

Subword modeling provides a significant boost to performance for POS only. Pre-trained implementations don't offer additional benefits. For NER, the best performance was obtained without sub-word modeling