**Part 1 Answers**

Roee esquire, ID 309840791

Yedidia Kfir, ID 209365188

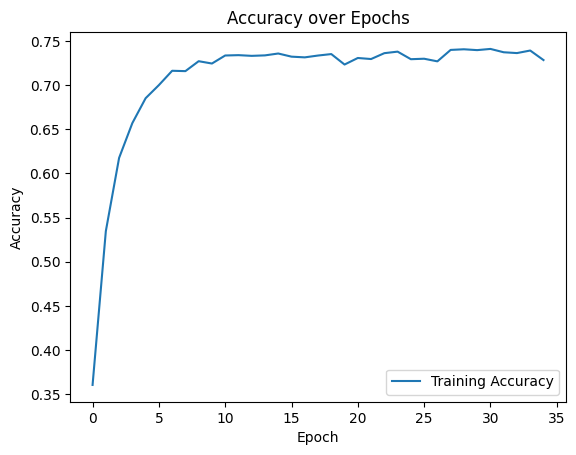
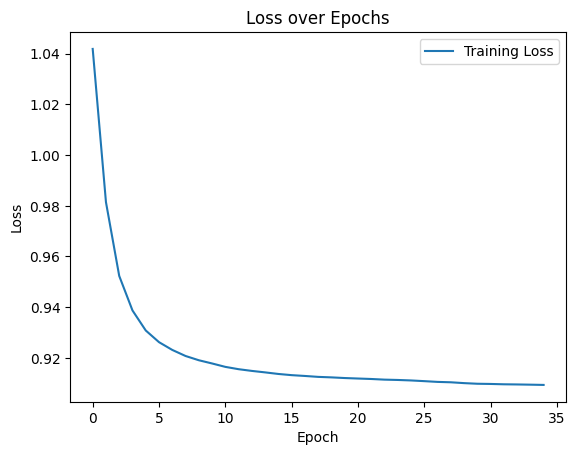
**Considerations**

1. Generally, for the prefixes and suffixes of word under length 3, we added the word as is to the prefix/suffix list.
2. For the padding words, the surrounding pseudo words, we added them to the prefix/suffix list for a general learned vector. Meaning we do not truncate them for their suffix, but rather keep a unique pseudo suffix and unique pseudo prefix for each pseudo word.
3. For words that their prefix/suffix is not found in train set, we use a general sub word embedding for these unknown words. This ‘unknown’ vector is learned.

**NER Results**

**without pre trained embeddings**

The best parameters we found was using hidden dimension 40 and 0.001 learning rate. This yielded 0.7409065472859541 accuracy rate on dev set after 30 epochs. See accuracy and loss function with hidden dimension 40:

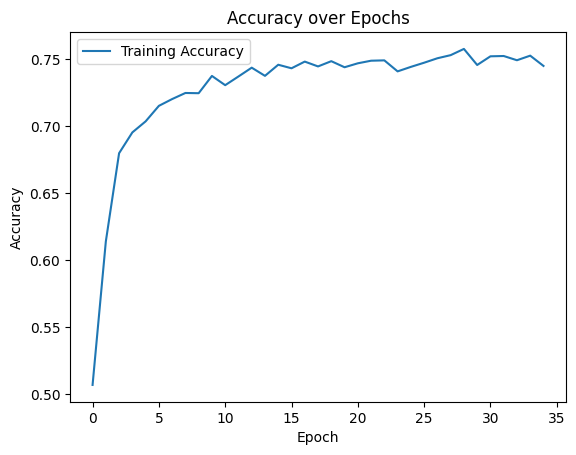
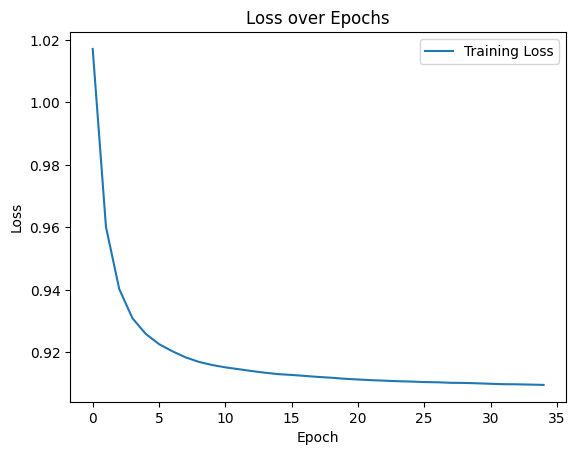


We also tested with hidden dimension 20, this resulted in peak accuracy of

0.7358390119250426 and hidden dimension 60 that peaked in accuracy of 0.7281620439636322.

**with pre trained embeddings**

The best parameters we found was using hidden dimension 40 and 0.001 learning rate. This yielded 0.7574368150301946 accuracy rate on dev set after 28 epochs. See accuracy and loss function with hidden dimension 40:



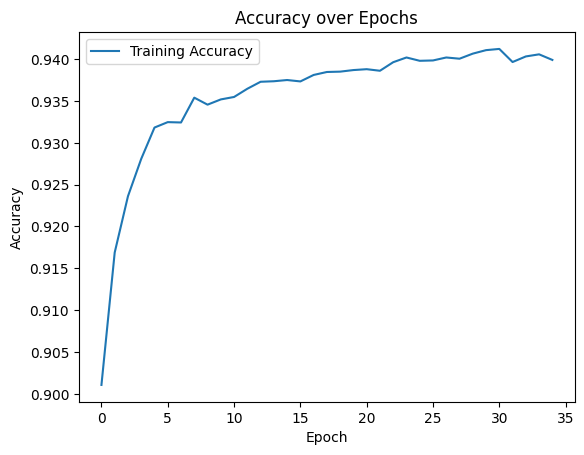
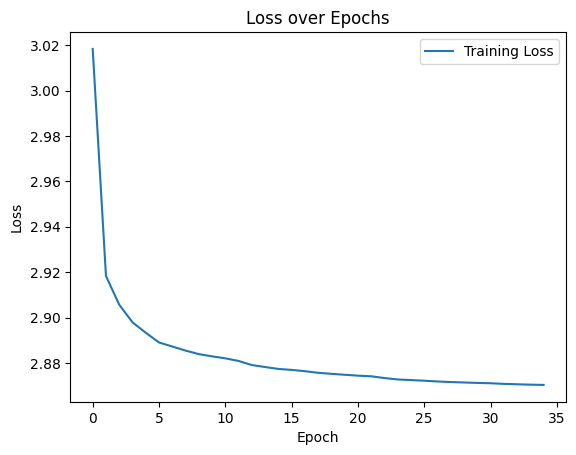
We also tested with hidden dimension 20, this resulted in peak accuracy of

0.7448766206608114 and hidden dimension 60 that peaked in accuracy of 0.7409456740442656.

**POS Results**

**without pre trained embeddings**

The best parameters we found was using hidden dimension 60 and 0.001 learning rate. This yielded 0.9412182338324041 accuracy rate on dev set after 30 epochs. See accuracy and loss function with hidden dimension 60:

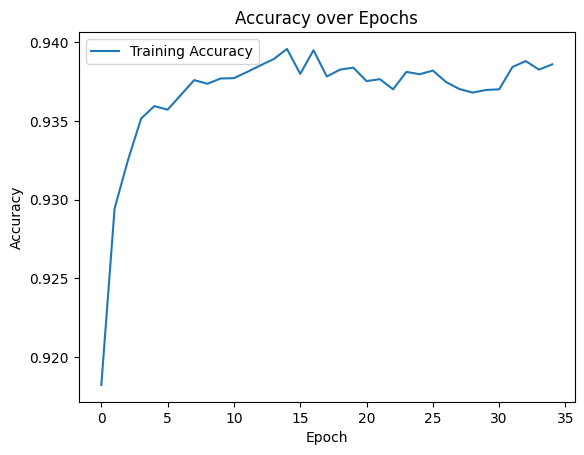
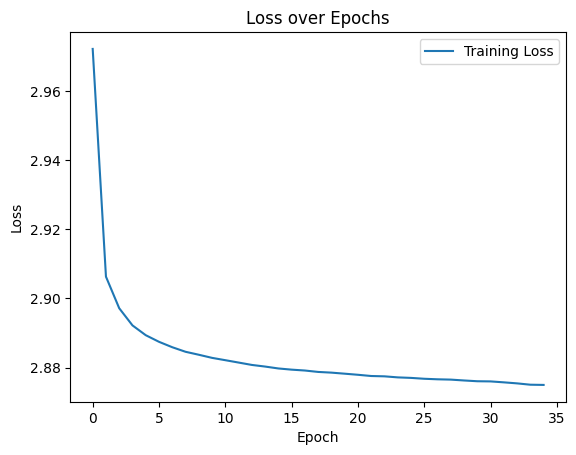


We also tested with hidden dimension 20, this resulted in peak accuracy of

0.9226753534214538 and hidden dimension 40 that peaked in accuracy of 0.9349328655849987.

**with pre trained vectors**

The best parameters we found was using hidden dimension 60 and 0.001 learning rate. This yielded 0.9395894673098206 accuracy rate on dev set after 14 epochs. See accuracy and loss function with hidden dimension 60:



We also tested with hidden dimension 20, this resulted in peak accuracy of

0.9272693102800226 and hidden dimension 40 that peaked in accuracy of 0.9266846248616593.

**Analysis**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Without pre trained embeddings, without sub word units | With pre trained embeddings, without sub word units | Without pre trained embeddings, with sub word units | With pre trained embeddings, with sub word units |
| **NER** | **0.709** | **0.72** | **0.74** | **0.757** |
| **POS** | **0.931** | **0.906** | **0.941** | **0.939** |

For NER it appears that the improvement is consistent with every addition of information. On the other hand, for POS the added information in the pre trained embeddings required extra information with the sub units in order to be used effectively. In POS there is big contribution to sub words unit since verbs and nouns have different suffixes and prefixes, for the most part, broadly speaking. We can deduce that the pre trained embeddings are more semantic in their information contribution, and are not granulated enough for usage in POS tagging. Sub words units contribute much more for POS than these embeddings.